THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

EVENT DETECTION IN SOCIAL NETWORKS

Sayan Unankard

Master of Science

A thesis submitted for the degree of Doctor of Philosophy at

The University of Queensland in 2015

School of Information Technology and Electrical Engineering

Abstract

Micro-blog services such as *Twitter* generate a large amount of messages carrying event information and users' opinions over a wide range of topics. The events discussed on social networks can be associated with topics, locations, and time periods. The events can be a variety, such as celebrities or political affairs, local social events, accidents, protests, or natural disasters. Messages are posted by users after they have experienced or witnessed the events happening in the real-world and they want to share their experiences immediately. People also express themselves spontaneously with respect to the social events in their social networks. Alternatively, policy-makers may want to know the feelings of users for a particular event to make informed decisions. With the increasing number of real-world events that are originated and discussed over social networks, event detection and tracking is becoming a compelling research issue. However, the traditional approaches to event detection and event tracking on large text streams are not applicable because of the following problems. First, they are not designed to deal with a large number of short and noisy messages. Second, social networks contain network structures such as friends, followers, replies, and *re-tweets*. Third, social network messages are associated with locations, which can be either senders' current locations or event locations. Fourth, each message is also associated with a timestamp. Messages often contain revealing and timely event information however, traditional text processing approaches assume documents are non-temporal. Moreover, given a particular time frame and a location the user is interested in, events that occurred in the given time frame from the chosen area are more valuable than others. Finding localized events has not been well studied yet.

The goals of this thesis are to: (1) identify subsistent problems and challenges in event detection and tracking in streaming micro-blog text, (2) design approaches for event detection and event tracking in social networks, (3) design approaches for sentiment analysis for given event topics, and (4) evaluate the proposed approaches in real-world streaming datasets.

In this thesis, our research is considered in three parts. Firstly, in order to detect emerging events from a large number of short and noisy messages, we propose an approach for the early detection of emerging hotspot events in social networks with location sensitivity. An algorithm is designed for slang conversion, synonym expansion and conceptual similarity to provide a rich semantic context for measuring message similarity to improve clustering results. We consider the message-mentioned locations for identifying the locations of events. In our approach, we identify strong correlations between user locations and event locations in detecting the hotspot emerging events. A sliding window manager is used to keep track of messages arriving in the system. The size of the sliding window is defined as the time interval. We evaluate our approach based on a real-world *Twitter* dataset. Our

experiments show that the proposed approach can effectively detect emerging events with respect to user's locations that have different granularities.

Secondly, for a long-running event like a nation-wide election which usually has fixed start and end times, users may want to monitor sub-events (i.e., hierarchically nested events that break down an event into more refined parts) such as the debate or campaign launch speech. Moreover, policy-makers may want to understand the feelings of users during the course of an election. We propose an approach for sub-event detection and sentiment analysis for a given long-running event. Given the user's initial event query, hierarchical clustering method is utilized for grouping the messages into sub-events. Lexicon-based approach is designed to detect user's opinion for specific entities. To evaluate our approach we present an approach to detect users' political preferences and predict the election results by incorporating sub-event detection and sentiment analysis at a state as well as a national level, as a case study. Our approach achieved better prediction results than the given baselines and comes close to the results of traditional polls. It might suggest that the discussions of sub-event topics that users had engaged in influenced their voting. Also, it can be seen that *Twitter* is able to reflect underlying trends in a political campaign.

Finally, with the variety of events discussed in micro-blog, people may be interested in understanding the whole situation of an event. For example, when natural disasters occur, people may start talking about what was happening and where an event was happening. Then, the effects of the event on the surroundings will be reported. Topics related to volunteer activities and cleaning up will be discussed at a later time. While all topics are related to the same situation, the clustering technique considers them as different events. In our final work, we introduce an invariant event tracking system, which is focused on analysing the continuous invariant events and their movements in a particular time period. We detect events by utilizing the Clique Percolation Method (CPM) community mining and track invariant events based on the relationships between communities. To demonstrate our approach, we use the *Twitter* messages related to the 2013 Australian federal election event with a given set of keywords search retrieved from the announced election day until the day after election day. The results show that our approach can capture the development of event for a given time period.

Declaration by Author

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

I have clearly stated the contribution of others to my thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of my research higher degree candidature and does not include a substantial part of work that has been submitted to qualify for the award of any other degree or diploma in any university or other tertiary institution. I have clearly stated which parts of my thesis, if any, have been submitted to qualify for another award.

I acknowledge that an electronic copy of my thesis must be lodged with the University Library and, subject to the policy and procedures of The University of Queensland, the thesis be made available for research and study in accordance with the Copyright Act 1968 unless a period of embargo has been approved by the Dean of the Graduate School.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate I have obtained copyright permission from the copyright holder to reproduce material in this thesis.

Publications during candidature

Journal papers:

- Sayan Unankard, Xue Li and Mohamed A. Sharaf. Emerging Event Detection in Social Networks with Location Sensitivity, *World Wide Web Journal*, pp. 1-25, 2014.

Conference papers:

- Sayan Unankard, Xue Li, Mohamed A. Sharaf, Jiang Zhong and Xueming Li. Predicting Elections from Social Networks based on Sub-Event Detection and Sentiment Analysis. In *Proceedings of Web Information Systems Engineering (WISE)*, Part II, pp. 1-16, 2014.

- Saeid Hosseini, Sayan Unankard, Xiaofang Zhou, Shazia Wasim Sadiq: Location Oriented Phrase Detection in Microblogs. In *Proceeding of Database Systems for Advanced Applications (DASFAA)*, pp. 495-509, 2014.

- Sayan Unankard, Xue Li and Mohamed A. Sharaf. Location-based Emerging Event Detection in Social Networks. In *Proceedings of Asia-Pacific Web Conference (APWeb)*, pp. 280-291, 2013. **[Best Student Paper Award]**

- Sayan Unankard, Ling Chen, Peng Li, Sen Wang, Zi Huang, Mohamed A. Sharaf, Xue Li. On the Prediction of Re-tweeting Activities in Social Networks A Report on WISE 2012 Challenge. In *Proceedings of Web Information Systems Engineering (WISE)*, pp. 744-754, 2012. **[Champion Data mining track]**

- Vinita Nahar, Sayan Unankard, Xue Li, and Chaoyi Pang. Sentiment Analysis for Effective Detection of Cyber Bullying. In *Proceedings of Asia-Pacific Web Conference (APWeb)*, pp. 767-774, 2012.

Posters:

- Sayan Unankard, Xue Li and Mohamed A. Sharaf. Sub-Event Detection and Sentiment Analysis in Social Networks, In *The 2014 Australasia Database Conference PhD School in Big Data*, The University of Queensland, Australia, 2014. **[Best Research Poster Awards - Joint First Prize]**

Publications included in this thesis

Sayan Unankard, Xue Li, Mohamed A. Sharaf, Jiang Zhong and Xueming Li. Predicting Elections from Social Networks based on Sub-Event Detection and Sentiment Analysis. In WISE, Part II, pp. 1-16, 2014. - incorporated as Chapter 4.

Contributor	Statement of contribution
Sayan Unankard (Candidate)	Designed algorithm and experiments (80%) Wrote the paper (80%)
Xue Li	Designed algorithm and experiments (20%) Revised the paper (20%)
Mohamed A. Sharaf	Proof reading for the paper
Jiang Zhong	Proof reading for the paper
Xueming Li	Proof reading for the paper

Sayan Unankard, Xue Li and Mohamed A. Sharaf. Emerging Event Detection in Social Networks with Location Sensitivity, World Wide Web Journal, pp. 1-25, 2014. - incorporated as Chapter 3.

Contributor	Statement of contribution
Sayan Unankard (Candidate)	Designed algorithm and experiments (80%) Wrote the paper (80%)
Xue Li	Designed algorithm and experiments (20%) Revised the paper (20%)
Mohamed A. Sharaf	Proof reading for the paper

Sayan Unankard, Xue Li and Mohamed A. Sharaf. Location-based Emerging Event Detection in Social Networks. In APWeb, pp. 280-291, 2013. - incorporated as Chapter 3.

Contributor	Statement of contribution
Sayan Unankard (Candidate)	Designed algorithm and experiments (70%) Wrote the paper (70%)
Xue Li	Designed algorithm and experiments (20%) Revised the paper (30%)
Mohamed A. Sharaf	Discussion of the algorithm design (10%) Proof reading for the paper

Sayan Unankard, Ling Chen, Peng Li, Sen Wang, Zi Huang, Mohamed A. Sharaf and Xue Li. On the Prediction of Re-tweeting Activities in Social Networks - a report of the WISE 2012 Challenge. In WISE, pp. 744-754, 2012. - incorporated as Chapter 1 and 2.

Contributor	Statement of contribution
Sayan Unankard (Candidate)	Conducted statistical studies (100%) Designed algorithm and experiments (25%) Wrote the paper (20%)
Ling Chen	Designed algorithm and experiments (25%) Wrote the paper (20%)
Peng Li	Designed algorithm and experiments (25%) Wrote the paper (20%)
Sen Wang	Designed algorithm and experiments (25%) Wrote the paper (20%)
Xue Li	Wrote and revised the paper (20%) Discussion and analysis of the algorithm design
Zi Huang	Proof reading for the paper
Mohamed A. Sharaf	Proof reading for the paper

Contributions by others to the thesis

For all the published research works included in this thesis, Associate Professor Xue Li, as my principle advisor, has provided very helpful insight in the overall as well as the technical details. He also assists with both the refinement of the idea and the pre-submission edition. For all of the research problems in this thesis, the principle advisor of the author, Associate Professor Xue Li, assisted in providing guidance for problem formulation, idea refinement as well as reviewing and polishing the presentation. My co-advisor, Dr Mohamed A. Sharaf, assisted in refinement of research problems and reviewing the presentation.

Ms Ling Chen, Mr Peng Li and Mr Sen Wang participated in conducting experiments and paper writing for our preliminary study incorporated in Chapter 2.

Statement of parts of the thesis submitted to qualify for the award of another degree

"None"

Acknowledgments

It is a humbling experience to acknowledge those people who have, mostly out of kindness, helped along the journey of my PhD. I am indebted to so many for encouragement and support.

I would like to express my special appreciation and thanks to my principal advisor, Associate Professor Xue Li; he has been a tremendous mentor for me. I would like to thank him for his expert advice and encouragement throughout this difficult research and for allowing me to grow as a research scientist. His guidance has made this a thoughtful and rewarding journey. I would also like to thank my associate advisor, Dr Mohamed Sharaf, for his comments and suggestions. For all committee members, especially Dr Helen Huang as the chair of committee and Professor Shazia Sadiq as the chair of examiners, I would like to thank you for all criticism and advice to perfect my thesis.

I must express my gratitude for the support and friendship provided by the other members of DKE group especially my roommates in room 78-636. I am indebted to them for their help and encouragement. To Professor Heng Tao Shen and other students, thank you very much for playing badminton with me to make me fit, healthy and get rid of stress at work.

To my friends, thank you for listening, offering me advice, and supporting me through this entire process. To my family, thank you for understanding me in all of my decisions to follow my dreams.

Finally, this thesis would have been impossible without the support of a Royal Thai Government scholarship. I must express my gratitude to the Thai Government not only for providing the funding which allowed me to undertake this research, but also for giving me the opportunity to attend conferences and meet so many interesting people.

Keywords

emerging event detection, location-based social networks, short text clustering, synonym expansion, conceptual similarity, sentiment analysis, invariant event, event tracking, micro-blogs, social network analysis

Australian and New Zealand Standard Research Classifications (ANZSRC)

ANZSRC code: 080107 Natural Language Processing, 40%
ANZSRC code: 080109 Pattern Recognition and Data Mining, 60%

Fields of Research (FoR) Classification

FoR code: 0801, Artificial Intelligence and Image Processing, 100%

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the present age, social networks have become the most popular way of communication for the current generation. The number of social network activities has increased dramatically, for example, information sharing, daily conversation and spreading news [50]. User-generated content (UGC) systems, like micro-blog services provides a wealth of current topics about real-world events which are discussed in social networks communities. Micro-blog like *Twitter* is being considered as a powerful means of communicating for people looking to share and exchange information on a wide variety of real-world events. *Twitter* is an on-line social networking service that enables users to send and read short 140-character messages called ***tweets***[1]. In 2014, the service rapidly gained worldwide popularity, with more than 500 million users who posted 500 million *tweets* per day. It has 284 million monthly active users. Moreover, 80 percent of *Twitter* active users are on mobile[2].

As *Twitter* has become the most popular way to share information about real-world events, sometimes an event can be reported even before the mainstream media. For instance, the explosions at the Boston Marathon 2013 and the death of the former British Prime Minister Margaret Thatcher in April 2013[3] were reported by social media first. The events can be a variety, ranging from popular, events concerning celebrities or political affairs, to local events, such as accidents, protests or natural disasters. In addition, social events can be anti-social, unlawful, or harmful to public security. For instance, in August 2011, rioters used instant messaging and social network services to arrange

[1] http://en.wikipedia.org/wiki/Twitter
[2] https://about.twitter.com/company
[3] http://www.guardian.co.uk/technology/2013/apr/23/twitter-first-source-investment-news

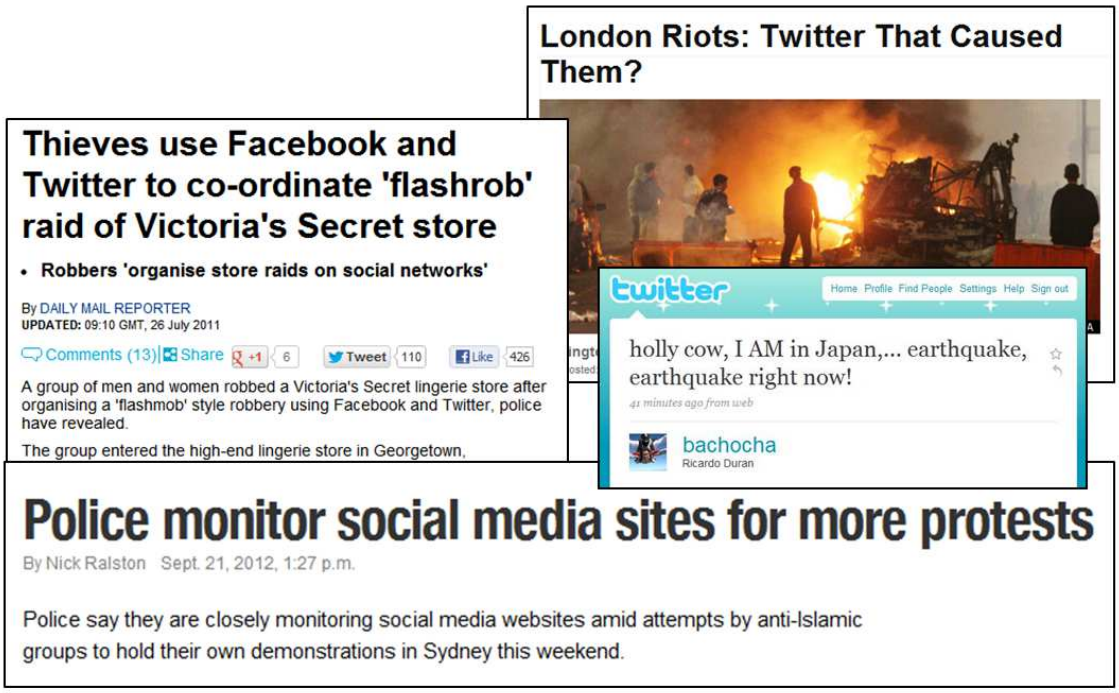


FIGURE 1.1: The use of social networking in spreading information.

meetings of agitated people across England.

In this chapter, we briefly introduce the research including motivation, background, main contributions and the organization of this thesis.

1.1 Motivation

The use of social networking in spreading information of emergent events (such as fires, bombings, natural disasters and disease outbreaks) and malicious incidents like the London riots as shown in Figure 1.1 has inspired a research challenge for the early detection of emerging events and the tracking of a particular event. Emerging events such as infectious diseases and cyberspace-initiated/plotted attacks/unrest need to be detected in their early stages. Emerging events like natural disasters may need to be reported in real-time when they are observed by people. However, with the large amount of short and noisy messages currently available on social networks, it is difficult to filter and sort-through posts manually as well as monitor emerging events as they are unfolding. Therefore, having a mechanism that can automatically perform this task in real-time would be very beneficial to governmental departments such as disaster-response and epidemic-prevention departments. In addition, there are a

number of event detection systems from social text streams. Currently, existing studies are considered on global events [96, 114, 123] and local events [112]. Moreover, there are several researches of emerging event detection [4, 12, 67, 87] which are focused on global events and some of studies are focused on identifying events of a particular type, such as earthquakes [91], news events [94], and swine flu [98]. However, this task cannot be achieved by classifying each message in real-time on the platform; the classes cannot be predefined because new events constantly appear in the social stream, and labelling *tweets* for training is not feasible as a result of the huge amount of messages posted.

For a particular event such as a natural disaster or a national election, people may want to track and understand the feeling of users in each topic. For example, during natural disasters or protests, the government may need to monitor the development of situations in order to make the right responses at the right times.

1.2 Background

Currently, a UGC system, such as micro-blog services, provides a wealth of online discussions about real-world events. Micro-blog, like *Twitter* is being considered as a powerful means of communication for people who are looking to share and exchange information over the social media. *Twitter* generates a large amount of messages carrying event information and users' opinions over a wide range of topics. The events discussed on social networks can be associated with topics, locations, and time periods. Messages are posted by users after they have experienced or witnessed the events happening in the real-world and they want to share their experiences immediately.

The fast information sharing on *Twitter* from millions of users all over the world leads to almost real-time reporting of events. This strong temporal nature of shared information allows for the detection of significant events in the data stream. Therefore, before we can successfully identify events in social networks, we must understand the scope of information that exhibits such trending behaviour, with the particular goal of characterizing and distinguishing between trends that reflect event information and trends that reflect other non-event content.

1.2.1 Micro-blog Data and Event

A micro-blog message is a short text message such as a *Twitter* message that is restricted to 140 characters and therefore is much more concise than a blog post. *Twitter* allows users to post short messages, or *tweets*, which are up to 140 characters. Users access *Twitter* in a number of different ways, including through the website interface, web services and third party applications. Importantly, a large fraction of the *Twitter* messages are posted from mobile devices and services, such as Short Message Service (SMS) messages. A user's messages are displayed as a stream on the user's *Twitter* home page. *Re-tweeting* is when a message is forwarded or re-posted via *Twitter* by users. Both *tweets* and *re-tweets* can be tracked to see which ones are most popular.

In terms of social connectivity, *Twitter* allows a user to follow any number of other users. The *Twitter* user can follow other users without requiring approval. Users can set their privacy preferences so that their updates are available only to each user's followers. By default, the posted messages are publicly available to anyone. In this work, we only consider messages posted publicly on *Twitter*.

Users can group posts together by topic or type by the use of *hashtags*; words or phrases prefixed with a "#" sign such as "#protest", "#uq" and "#earthquake". Similarly, the "@" sign followed by a username is used for mentioning or replying to other users. To re-post a message from another *Twitter* user and share it with one's own followers, a user can click the *re-tweet* button within the message. *Re-tweet* commonly uses the "RT @username" text as prefix to credit the original (or previous) author. Examples of *re-tweet* messages are as follows:

RT @SummersAnne: "The first group of asylum seekers have left Christmas Island and are due to land in PNG this morning http://t.co/pkhpAtsm"

RT @nickpmclaren: "Where the major parties stand on mining in the lead up to the 2013 poll. #coal #gas #CSG #shale #uranium http://t.co/BUv"

As there is a 140-character limit, shorthand notation and slang are commonly used in the message content. It has also increased the usage of URL shortening services such as bit.ly and goo.gl. As the number of posts has grown, it is clear that *Twitter* is an information system that provides a real-time trending topic of the interests of its users, as well as their attention. A trend on *Twitter* consists of one or more terms and a time period, such that the volume of messages posted for the terms in the time

Australia Trends · Change	Thailand Trends · Change
#RemembranceDay	#หัวข้อกระทู้pantipในอนาคต
#LestWeForget	#300DaysWithGOT7
#RuinA90sBand	#LeaderJB
#HappyBirthdayJacob	#HISUHYUN
#BLANKSPACEMUSICVIDEO	#เพราะผู้หญิงไม่ได้เหมือนกันทุกคน
Ben Barba	Taylor Swift - Blank Space
Wayne Goss	JustinBieber
Christmas	Pocky Day
Larry	Thailand
Melbourne	Interstellar

FIGURE 1.2: The screenshot of *trending topics* on *Twitter* in two different locations.

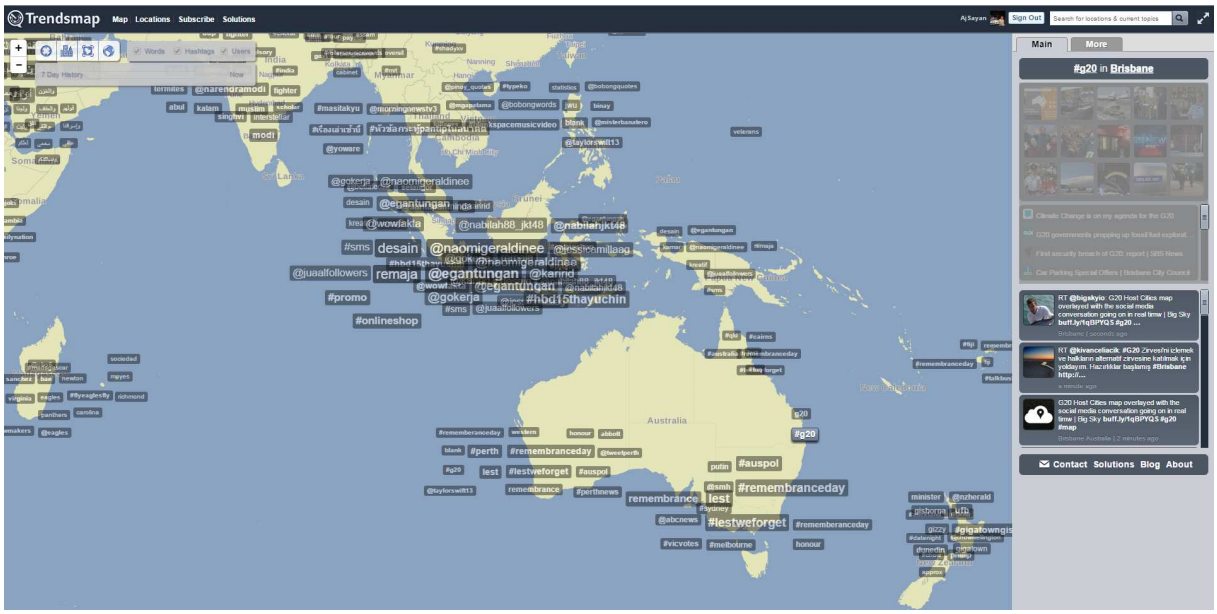


FIGURE 1.3: The screenshot of *TrendsMap*.

period exceeds some expected level of activity[4]. An example of *Twitter* trend is shown in Figure 1.2. There are many commercial applications designed to discover social network trends. For example, there is *TrendsMap*[5] which is a real-time mapping of *Twitter* trends (popular keywords) across the world. The screenshot of *TrendsMap* is shown in Figure 1.3. In fact, the majority of trend topics can be considered headline news or persistent news [50]. However, trend topics are not all real-world events. Also, using only one keyword to represent a trend topic is unable to fully understand what happening.

[4]http://en.wikipedia.org/wiki/Twitter#Trending_topics
[5]http://trendsmap.com

The rising popularity of online social network services has motivated research to understand their characteristics [9, 115]. Java et al. conducted a preliminary analysis of *Twitter* in 2007 [42]. Their dataset covered about 76,000 users and 1,000,000 messages. They found user clusters based on user intention to topics by clique percolation methods. Krishnamurthy et al. also analysed the user characteristics by examining the relationships between the number of followers and the number of followings [48]. Zhao and Rosson [122] qualitatively investigate the motivation of using *Twitter* at work. Jansen et al. in [40, 41] conducted a preliminary analysis of word-of-mouth branding in *Twitter*.

In this thesis, our *Twitter* data consists of *tweet* ID, creating time, user ID, user location, *tweet* location, text content and the message-mentioned locations. A *tweet* location is known when a user posts the message using a smart phone. An example of *Twitter* data is shown in Table 1.1. In this thesis, we define an event as follows:

Definition 1.1. *An "event" is something that occurs in a certain place during a particular interval of time*[6].

An event location is a place where the event will happen or is happening while a topic location is a place that is included in the topic content. In the real-world, an event location can also be a topic location, or they can be different from each other. In this research we only consider event locations. The topic locations are not considered exclusively, e.g., an "earthquake" location is both the topic as well as the event location. Therefore, we define an emerging event and a hotspot event as follows:

Definition 1.2. *An "emerging event" is an event that has significantly increased in the number of messages but rarely has been posted in the past.*

Definition 1.3. *A "hotspot event" is an event where there is a strong association between event location and user location.*

User location is the location where the message is sent from. Message-referred location is the location mentioned in the message. It could be the event location or other locations referred to within the messages.

[6] http://dictionary.reference.com/browse/event?s=t

TABLE 1.1: An example of *Twitter* data.

tweet id	created time	user id	user location	tweet location	content
1	2012-08-21 08:26:45	2285xxx	Brisbane, Australia	-27.480835, 153.030392	STOP THE CUTS: march on Qld Parliament. Meet 5pm at **King George Square, Brisbane** this Thursday 23rd
2	2012-08-21 08:32:55	1957xxx	Queensland	-	Stop the Cuts Rally this afternoon 5pm **King George**. **Square** I'll be there. http://t.co/UqH3EMOl
3	2012-08-21 08:33:10	3017xxx	Queensland	-	RT @MEANIEgrrl: STOP THE CUTS: march on Qld Parliament. Meet 5pm at **King George Square**, Brisbane this Thursday 23rd

In order to understand where the locations of messages are found in social media, we conducted statistical studies. Firstly, we tried to understand the availability of the locations within the micro-blogs. We used *Twitter API*[7] to crawl the micro-blog datasets. *Dataset 1* is crawled from the messages sent by users around Brisbane Australia, from the dates 17 March 2012 to 25 March 2012 with 219,933 messages, while *Dataset 2* is crawled from the messages sent by users around the USA, from the dates 21 June 2012 and 27 June 2012, with 196,834 messages. The statistical information of locations from the two datasets is shown in Figure 1.4 and Table 1.2.

As we can see from *Datasets 1 and 2*, the user locations are divided into three types i.e., *geo-tagged* locations, user profile locations, and those implied by IP addresses (first three rows in the Table 1.2). The majority of user locations comes from user profiles; approximately 77% of *Dataset 1* and 67% of *Dataset 2*. Also from *Datasets 1 and 2*, we can see that the message-mentioned locations are only available within a small proportion of the micro-blogs and appear one or two times on average in the micro-blog messages. Our most important observation on the locations of micro-blogs is that most of the micro-blog messages contain only one geographical location while messages that contain more than one geographical location constitute approximately 4% and 7% of *Dataset 1* and *Dataset 2* respectively. When a location is mentioned in the micro-blog message, it can be either an event location or a topic location.

[7] https://dev.twitter.com/docs/api/1.1

FIGURE 1.4: A bar chart of micro-blog message's location.

In Table 1.3, we also tried to understand what the locations are used for in the messages' contents. *Dataset 3* is downloaded from G. R. Boynton, University of Iowa and consists of two events: USA H1N1[8] and the Indonesia Earthquake[9]. *Dataset 4* is crawled by using hashtag *"#qldvote"* for Queensland Election 2012. As we can see from *Datasets 3 and 4*, the message-mentioned locations are mostly the event locations with the confidence being higher than 92 percent. The message-mentioned location can appear more than one time in the message.

Examples are shown as follows:

Message 1: *"Earthquake hits* ***Indonesia*** *now!!* ***#indonesia*** *#quake"*

Message 2: *"tsunami in* ***samoa****, earthquake in* ***indonesia****...."*

Our most important observation on the message-mentioned locations is that the more frequent a message-mentioned location, the more likely it is the event location. Therefore, we use message-mentioned locations to identify event locations in our work.

[8]http://ir.uiowa.edu/polisci_nmp/2/
[9]http://ir.uiowa.edu/polisci_nmp/5/

TABLE 1.2: Statistics of message locations.

List	Dataset1		Dataset2	
	#msgs	%	#msgs	%
Messages with geo-tagged location	3,121	1.42	5,055	2.57
Messages with user profile location	170,334	77.45	132,661	67.40
Messages with IP address-based location	46,478	21.13	59,118	30.03
Messages that contain geographical location in contents	16,875	7.67	19,529	9.92
Messages that contain geographical location in contents (more than 1 location)	744	4.41	1,337	6.85

TABLE 1.3: Analytics of message-mentioned locations.

Data set	**Event**	**Total No. of msgs**	**Msgs contain mentioned location**		**#of time that event location occur**	**#of times that other locations occur**	**Confidence of location mentioned in the content as event location**
			#of msgs	**%**			
3	H1N1	958	79	8.25	73	6	92.41%
	Earthquake	801	798	99.66	846	37	100%
4	QLD Election 2012	629	87	13.83	86	1	98.85%

1.2.2 Message Propagation in Social Networks

Understanding how information spreads through large user communities is also important. The success of online social networks opens a new problem of large-scale information diffusion. Topic propagation in blogspace [30], favourite photo marking in a social photo sharing services [13], fanning in Facebook [101], and meme tracking in news cycles [54] all report on large-scale information diffusion online. There are research works that focus on characterising information diffusion networks in *Twitter* [30, 50, 53, 117, 10, 25, 119, 103]. In [10], Boyd et al. studied the various aspects of *re-tweeting*. They conducted interviews with *Twitter* users and investigated how people *re-tweet*, what they *re-tweet* and the reasons why they *re-tweet*. Galuba et al. in [25] focused on the *URL* propagation

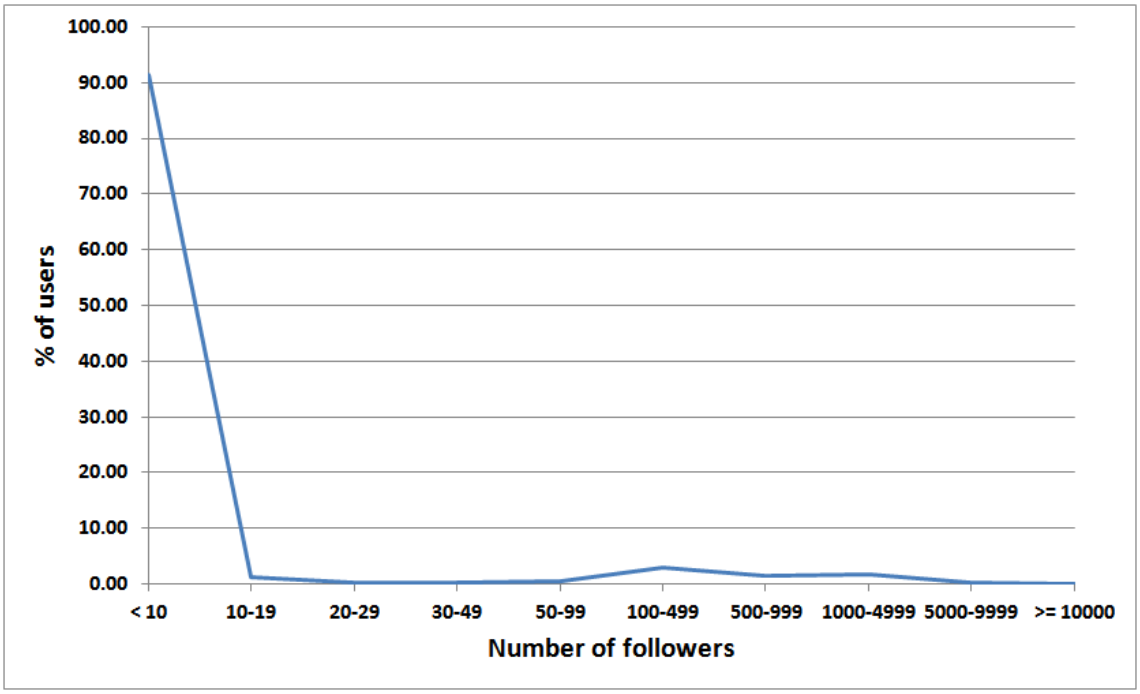


FIGURE 1.5: User distribution based on the numbers of followers.

via *re-tweets*. Yang et al. in [119] studied the *re-tweeting* behaviours. Tazidou in [103] investigated real-time analysis methods on social media with a focus on information diffusion.

In a micro-blog service, information can be spread in the form of *re-tweeting*. *Re-tweeting* is an action to re-post or forward another user's message on the social networking website *Twitter*[10]. People may want to *re-tweet* in order to enlarge or spread the messages to new audiences, to publicly agree with someone and to build relationships with others [10]. In addition, we can track the flow of information in micro-blog services via *re-tweeting* activities because they denote situations when a message is important or interesting enough to share with your own followers [121].

In order to understand the *re-tweeting* activities, we use the datasets crawled by the *WISE 2012* Challenge[11] from *Sina Weibo*, which is the largest Chinese micro-blogging site similar to *Twitter*. In *Sina Weibo*, the *re-tweet* mechanism is different from *Twitter*'s. In *Twitter*, users can only *re-tweet* a *tweet* without modifying the original *tweet*. However, in *Sina Weibo*, users can modify or add information from other users' in the *re-tweeting* path in their own *re-tweet*.

The dataset that will be used in this study contains two sets of files. Firstly, the followship network dataset includes the following network of users based on user *IDs*. Secondly, the micro-blog dataset

[10] http://en.wiktionary.org/wiki/retweet
[11] http://www.wise2012.cs.ucy.ac.cy/challenge.html

TABLE 1.4: The 12 event categories in *WISE 2012* dataset (I)

Category	Event
Natural Disaster	Earthquake of Yunnan Yingjiang
	Japan Earthquake
	Yushu earthquake
	Zhouqu landslide
Product Release	iPhone 4s release
	Windows Phone release
	Motorola was acquisitions by Google
	Xiaomi release
Sports	Yao Ming retirement
	Spain Series A League
	Li Na win French Open in tennis
Famous people	The death of Muammar Gaddafi
	The death of Steve Jobs
	Family violence of Li Yang
	Tang Jun educatioin qualification fake
	The death of Kim Jongil
	The death of Osama Bin Laden
Social problem	Anshun incident
	China Petro chemical Co. Ltd.
	Foxconn worker falls to death
	Guo Meimei
	Incident of self-burning at Yancheng, Jangsu
	Shanghai government's urban management officers attack migrant workers in 2011
	Yao Jiaxin murder case
	Yihuang self-immolation incident
	The death of Wang Yue
	Case of running fast car in Heibei University
Public Security	Bohai bay oil spill
	Foxconn bombing in Chengdu
	Fuzhou bombings
	Shanxi
Protests	Chaozhou riot
	Mass suicide at Nanchang Bridge
	Protests of Wukan
	Qianxi riot
	Zhili disobey tax official violent

includes basic information about *tweets* (*posted time, user ID, messages ID*), mentions (i.e., user *IDs* appearing in messages), *re-tweet* paths, and whether they contain links. User *IDs* and message *IDs* are anonymized. Content of *tweets* are removed, based on *Sina Weibo*'s Terms of Services. Some *tweets* are annotated with events. For each event, the terms that are used to identify the event and a

TABLE 1.5: The 12 event categories in *WISE 2012* dataset (II)

Category	Event
Development Projects	Line 10 of Shanghai-Metro pileup
	Shenzhou-8 launch successfully
	Tiangong-1 launch successfully
Economy	House prices
	Individual income tax threshold rise up to 3500
Human right	Qian Yunhui
	Deng Yujiao incident
Accident	Gansu school bus crash
	Wenzhou train collision
Crime	Chongqing gang trials

TABLE 1.6: Number of original messages re-tweeted in 30 days.

Number of re-tweets	Original messages		Annotated with events	
	#messages	%	#messages	%
< 10	42,551,891	94.749	882,191	2.073
10-99	2,171,214	4.835	65,809	3.031
100-499	173,803	0.387	5,464	3.144
500-999	10,283	0.023	400	3.890
1,000-4,999	2,838	0.006	158	5.567
5,000-9,999	26	0.00006	2	7.692
$\geq$10,000	11	0.00002	1	9.091
Total	44,910,066	100.00	954,025	2.124

link to a *Wikipedia*[12] page containing descriptions of the event are given.

In the dataset, the given original *tweets* are annotated with some social events together with their corresponding keyword lists. It is difficult to automatically group events into different categories because some events are simply labelled by personal names or by location names. Moreover, their relevant keyword lists are arbitrary and do not show clear contextual information between the keyword list and the event title. To solve this problem, we manually divide the *WISE 2012* provided 46 events that have links to *Wikipedia* pages, into 12 categories such as Natural Disaster, Celebrities, Product Release, Sports, etc. The 12 event categories are shown in Table 1.4 and 1.5.

For the purpose of this study, 369 million messages and 68 million user profiles were extracted. The sizes of the followship dataset and the micro-blog dataset are 12.8 GB and 64.8 GB, respectively.

[12]http://wikipedia.org

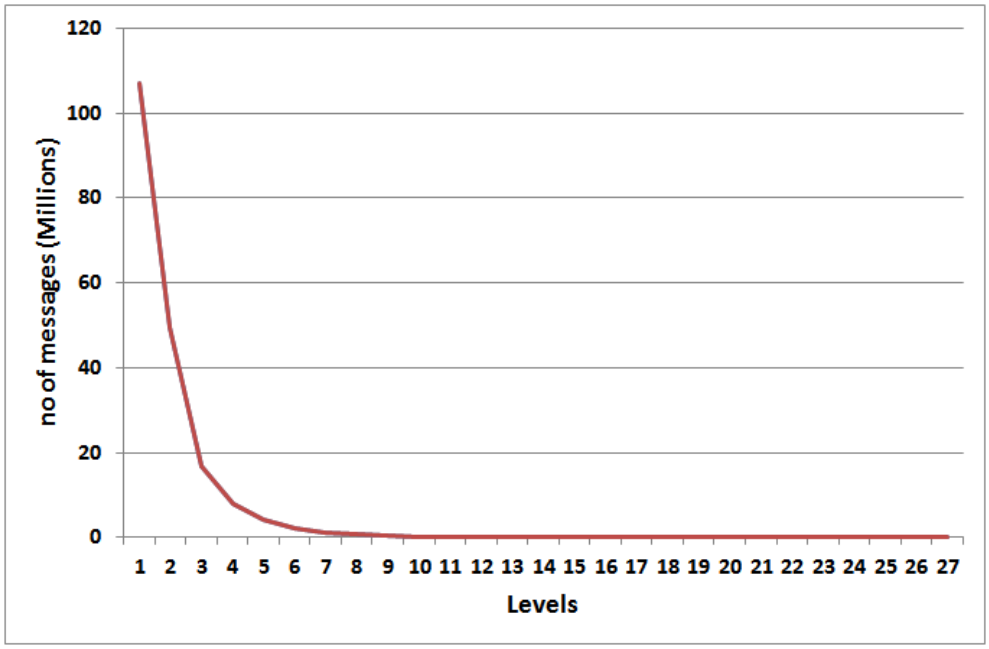


FIGURE 1.6: Number of re-tweets in each level.

It should be noted that the dataset is not complete but it is sufficiently large to predict the *re-tweeting* behaviour of users on *Sina Weibo*.

TABLE 1.7: Number of re-tweets in 10 levels within 30 days.

Level	Number of re-tweets	%
1	107,025,967	56.056
2	49,401,724	25.874
3	16,934,845	8.869
4	8,045,285	4.213
5	4,196,992	2.198
6	2,315,732	1.212
7	1,294,638	0.678
8	746,494	0.390
9	428,158	0.224
10	240,606	0.126

In preparation for this thesis, we further collected some statistical information for a better understanding of the available datasets. In particular, for the followship dataset (i.e., the "who is following whom" relationship), we found that the majority of users have less than 10 followers (approximately

91%) as shown in Figure 1.5. Additionally, for the micro-blog dataset (i.e., whose *tweets* are *re-tweeted* by whom), we ranked the distribution of the original *tweets* based on how many *re-tweets* they received within 30 days as shown in Table 1.6. The table also shows the subsets of *tweets* that have been annotated with events. As the table shows, approximately 95% of the original *tweets* were *re-tweeted* less than 10 times, of which approximately 2% were annotated with events. In addition, most original *tweets* were *re-tweeted* in three levels within 30 days (approximately 91%) as shown in Table 1.7 and Figure 1.6.

In order to understand the *re-tweet* activity, we also studied the *re-tweet* activity by day of the week and time of the day. We selected original *tweets* associated with popular events which have the number of *re-tweets* being more than 100 for our study (i.e., 6,934 messages). In Figure 1.7, the graph shows the number of *re-tweets* per day of week. Based on a sample of tweets, Monday is the most popular day for *re-tweet* activity, followed by Tuesday and Friday. Figure 1.8 shows the number of *re-tweets* per hour of the day. As can be seen, the most *re-tweet* activity during the day happens from 10 a.m. to 12 p.m. The proposed approaches to predict the volume of future re-tweets and possible views can be seen in [106].

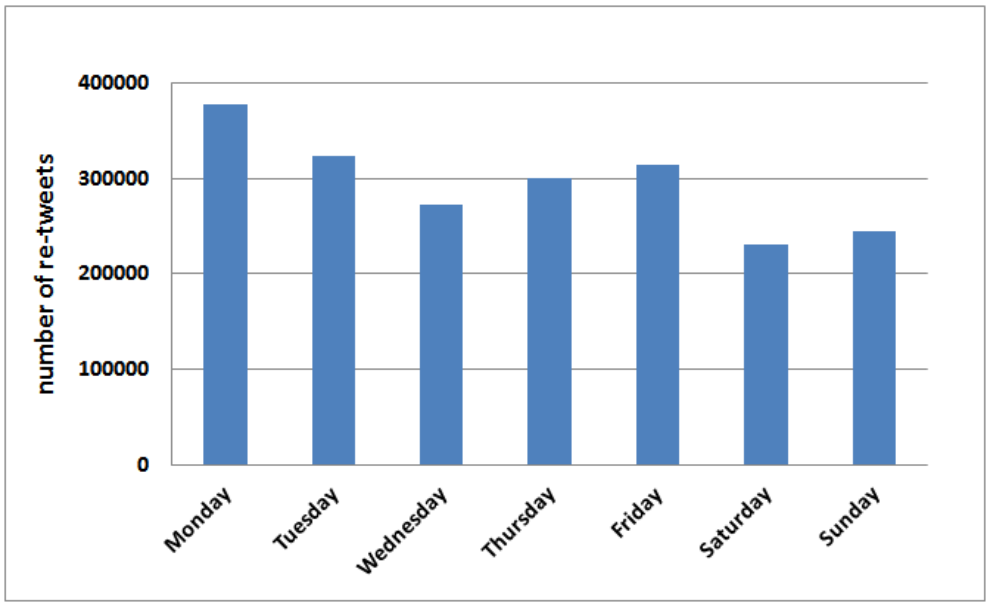


FIGURE 1.7: Re-tweet activity by day of the week.

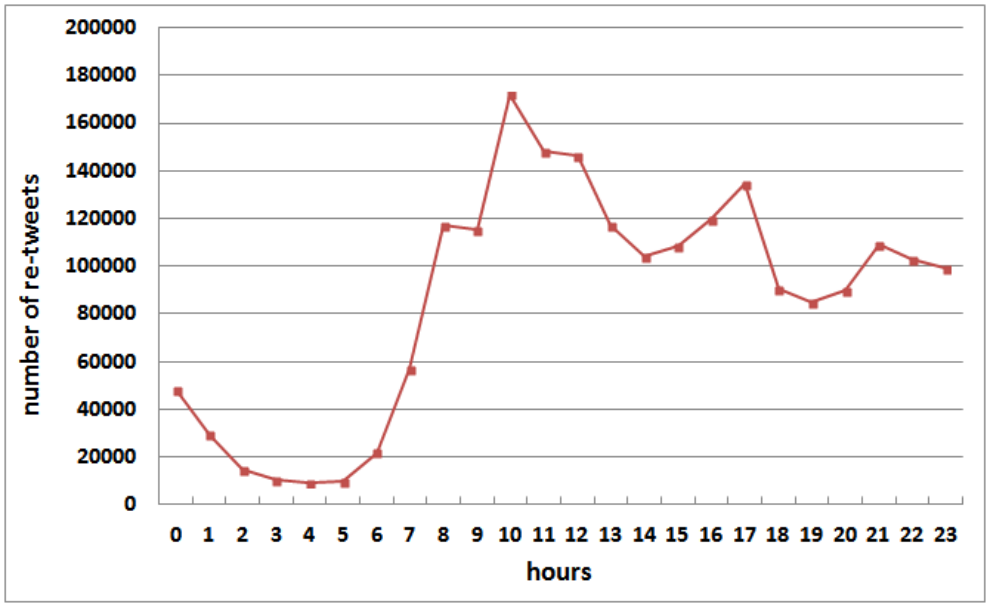


FIGURE 1.8: Re-tweet activity by time of the day.

1.3 Research Challenges

The problem that we address in this thesis are: (1) how to identify and track events from a given set of micro-blog messages, and (2) how to extract the feelings of users for a particular event topic. The topic of event detection and event tracking task were discussed in existing works [3, 118, 49]. In their works, events are identified from a stream of news such as newswire and radio broadcast transcripts. These news articles contain grammatical, syntactical, and stylistic standards where the writing used has a different style from that used in the micro-blog messages. Micro-blog messages like *Twitter* usually contain the form of a short description or keyword tags. Abbreviations are also widely used in a message. Moreover, the messages are often noisy.

The traditional approaches are not designed to deal with a large number of short and noisy messages. Micro-blog messages are correlated with network structures such as friends, followers, replies, and *re-tweets*. Moreover, social network messages are associated with timestamps and locations, which can be either senders' current locations or event locations, but traditional approaches assume that text documents are non-temporal. Given a particular time frame and location that an user is interested in, events that occurred in the given time frame from the chosen area are more valuable than others. Finding localized events has not been well studied yet. More importantly, with a large number and variety of events discussed on social networks, we do not know the number of events in advance.

Traditional data mining approaches such as clustering face with the new challenges for dealing with dynamic topics or events.

For sentiment analysis, the challenges are that the given topic may have subtopics which need to make sentiment analysis fine-grained and the cyber-sarcasm which makes sentiment analysis more difficult.

1.4 Research Goals

This section discusses the research goals of this thesis. At the highest level, our goal is to identify subsistent problems and challenges in event detection and tracking in social networks, paying attention to event detection and tracking in micro-blog services (i.e., *Twitter*). The goals of this thesis are as follows:

- A key goal is to design frameworks and develop effective solutions for event detection and tracking in streaming micro-blog text. Our approaches need to tackle with a large amount of short and noisy messages, social network structures, location and time associated with the messages.

- The second key goal is to design framework for sentiment analysis for a given event topic. Our approach focuses on a lexicon-based approach, thus we need to deal with lexicon dictionary, aspect sentiment analysis and sarcasm identification.

- The third key goal is to evaluate the proposed approaches in real-world streaming datasets. A real-world *Twitter* datasets will be used to evaluate our proposed approaches.

1.5 Main Contributions

Based on the research problems discussed and the challenges identified, we make the following contributions in this thesis towards event detection and tracking in social networks. The contributions of this thesis are as follows:

1.5.1 Location-based Emerging Event Detection in Social Networks

The problem that we address in this work is how to identify emerging events with location sensitivity from a given set of micro-blog messages. We consider a set of messages where each message is associated with an event. However, due to the characteristics of micro-blog messages, several issues are listed as follows:

- People share various types of content such as conversation topics, advertisements, events, opinions, and others. Our goal is to detect only emerging hotspot events that are happening in a particular area within a given time period.
- The weighting scheme of micro-blog messages should differ from traditional methods because the micro-blog message is very short and often does not provide sufficient information. Abbreviations are also widely used in a message.
- With the large range of events discussed on social networks, we do not know the number of events in advance. Traditional clustering methods like the K-Means technique should determine the fixed number of clusters; however, it is unsuitable for the real-world system when dealing with dynamic topics or events.

The main contributions are summarized as follows:

- An effective method to detect the emerging hotspot events is proposed.
- An approach to correlate user location with event location in order to establish a strong correlation between them is proposed to identify hotspot emerging events.
- An algorithm is designed for slang conversion, synonym expansion and conceptual similarity to provide a rich semantic context for measuring message similarity to improve clustering results.
- An effective evaluation for event detection on a real-world *Twitter* dataset with different granularities of locations is performed.

This research [107] was published in the proceedings of the 15th Asia-Pacific Web Conference (APWeb) 2013 and the extended version [108] was published in the World Wide Web Journal (WWWJ), 2014.

1.5.2 Sub-Event Detection and Sentiment Analysis in Social Networks

In this work, we present an approach by incorporating sub-event detection and sentiment analysis to analyse as well as visualise political preferences revealed by those social network users. To evaluate our methodology we utilized our approach to predict the election results at a state as well as a national level as a case study. The main contributions of this work are as follows:

- We design an approach to forecast the vote of a sample user based on the analysis of his/her micro-blog messages and count the votes of users to predict the election results.
- Sub-event detection and sentiment analysis are incorporated to predict the vote of users as different levels of sub-events that the user engaged in the discussions will affect the prediction results.
- We evaluate our proposed approach with real-world *Twitter* data posted by Australia-based users during the 2013 Australian federal election.

This research [109] was published in the proceedings of the 15th International Conference on Web Information System Engineering (WISE) 2014.

1.5.3 Invariant Event Tracking in Social Networks

In this work, we introduce a new concept called invariant event tracking. Invariant event tracking is important for analysing the overall situation of a particular event on social networks. For example, during a natural disaster, the government may need to analyse the development of situations in order to make the right responses at the right times. For a longer-running event such as a government election, people may wish to track the event with-respect-to multiple issues such as campaign-launch speeches and a number of open TV debates under different topics, in order to cast their votes. However, general micro-blog searches for given keywords return large amounts of messages that are not grouped or organized in any meaningful way. It is difficult for people to comprehend a large number of messages in a chronological order and to monitor an event as it unfolds. The main contributions of this work are twofold:

- An effective approach of tracking invariant events is proposed by incorporating CPM community mining and community evolution discovery techniques.
- We have implemented an invariant event tracking system which provides users with an overview of the development of an event. The system supports event tracking by allowing users to specify the time period in order to visualize the words consequently appearing and disappearing over time.

1.6 Thesis Organization

The remainder of this thesis is organised as follows. In Chapter 2, we investigate the literatures related to the research topics in this thesis. Chapter 3 focuses on an event detection method in social networks (i.e., micro-blog service). We present an approach for the early detection of emerging hotspot events in social networks with location sensitivity. Chapter 4 presents sub-event detection and sentiment analysis approaches for a particular event. Monitoring a particular event would be of benefit to everyone who wants to understand the story of an event and the current opinions on each sub-event topic. In Chapter 5, we propose an invariant event tracking approach. We use this system to track an event based on micro-blog messages and to monitor the topic changes over time for an event that is rendered to the system as a set of keywords. Finally, the conclusions and future research directions of this thesis are summarised in Chapter 6.

Chapter 2

Literature Review

The fast information sharing on social networks from millions of users all over the world leads to the almost real-time reporting of events. Event information can be spread in a social network in different ways. Before we can successfully identify events in social networks, we need to understand how event information is spread and how to measure the propagation behaviour of a certain event in a micro-blog social network. In a micro-blog service, information can be spread in the form of *re-tweeting*. People may want to *re-tweet* in order to enlarge or spread the messages to new audiences, to publicly agree with someone and to build relationships with others [10]. In this chapter, first we review research works on message propagation in social networks and present our preliminary work to predict the *re-tweet* activity of any given message for a particular event. Second, we describe some of the related works on detecting events in social networks, paying attention to event detection in micro-blog services (i.e., *Twitter*). Third, we review research works on short text clustering especially on-line clustering with text streams. Fourth, some research on event tracking will be presented. Finally, we describe related works on sentiment analysis in social networks.

2.1 Information Propagation in Social Networks

2.1.1 *Re-tweet* Message Propagation

The prediction of message propagation is one of the major challenges in understanding the behaviours of social network users. One interesting problem is the study on the *re-tweeting* behaviours from an

information diffusion perspective. Most works have focused on *Twitter*, a popular micro-blogging site. There are research works that focus on characterising information diffusion networks in *Twitter* [10, 121, 82, 80]. Insightful studies on *re-tweeting* behaviours can be described as follows.

In [10], Boyd et al. study the various aspects of *re-tweeting*. They conducted interviews with *Twitter* users and investigated how people *re-tweet*, what they *re-tweet* and the reasons why they *re-tweet*. Letierce et al. in [55] survey how researchers use *Twitter* to spread scientific messages. Galuba et al. in [25] focus on the *URL* propagation via *re-tweets*. The researchers tracked 15 million *URLs* exchanged among 2.7 million users over a 300 hour period. In [100], Suh et al. gathered content and contextual features from *Twitter* and identified factors that impact *re-tweeting*. The researchers found that *URLs* and *hashtags* have strong relationships with *re-tweetability*. The numbers of followers and followees are also identified as important factors. However, neither of them attempted to predict whether a given message was to be *re-tweeted*.

Zaman et al. in [121] adapt a probabilistic collaborative filtering model called *Matchbox* [99] to predict information spreading in *Twitter*, based on different features such as *tweeter* information (i.e., user who posted original message), *re-tweeter* information and the *tweet* content with one hour of messages. In [120], Yang et al. propose a factor graph model based on users' *re-tweeting* history. The researchers define 22 features in the training process such as the users' history preferences, messages' contents, information of the trace, and the time delay.

Petrovic et al. in [82] built a time-sensitive model based on the passive aggressive algorithm (PA) to automatically predict *re-tweet* activities. The features are divided into two sets (i.e., social features and *tweet* features). Social features consist of the number of followers, friends, statuses, favourites, number of times the user was listed, whether the user is verified, and whether the user's language is English. On the other hand, *tweet* features included the number of *hashtags*, mentions, *URLs*, trending words, length of the message, novelty, whether the message is a reply, and the actual words in the message. Hong et al. in [33] train a binary classifier to predict if a message will be *re-tweeted* or not and a multi-class classifier based on logistic regression to predict the volume of *re-tweets* for a given message. For the multi-class classification, they use four class labels (0: no *re-tweet*, 1: *re-tweets* less than 100, 2: *re-tweets* less than 10000, and 3: *re-tweets* more than 10000).

In [80], Peng et al. modelled the *re-tweeting* activities by using conditional random fields with

three types of features, namely content influence, network influence and temporal decay factor. Naveed et al. in [74] argue that the *tweet* content is the key for *re-tweeting* prediction. The researchers use logistic regression to compute *re-tweet* likelihood based on various interesting content features such as positive/negative emotion, and exclamation/question marks. The high-level features are formed by associating *tweets* to topics and by determining the sentiments of a *tweet*. Recently, predicting the future popularity trend of news events in micro-blogging platforms is proposed by Gupta et al. in [32]. Regression classification and hybrid approaches are studied using a large set of popularity, ratios, *URLs*, social and event features.

2.1.2 Predicting of *Re-tweeting* Activities

In our preliminary work, we aim to predict the propagation behaviour of any given short message (i.e., *tweet*) within a period of time. This is captured by measuring and predicting two metrics, namely: 1) the number of *re-tweets*, and 2) the number of possible views. The number of possible views of one *re-tweet* activity is defined as the number of followers of the user who conduct the *re-tweet* action. The number of possible views of an original *tweet* is defined as the sum of all possible view numbers of *re-tweet* actions. We manually divide the *WISE 2012* provided 46 events that have links to *Wikipedia* pages into 12 categories such as Natural Disaster, Celebrities, Product Release, Sports, etc. The 12 event categories are shown in Table 1.4 and 1.5 in Section 1.2.2.

Based on the given datasets, together with our statistical information presented in Section 1.2.2, we make the following assumptions:

- The given *tweets* to be predicted are original *tweets*.
- The events associated with the given *tweets* are popular events.
- An event category is a group of similar events (manually grouped).
- The more popular the event category is, the more likely the *tweet* will be *re-tweeted* by a user.
- Similar events have similar *re-tweet* patterns.
- A user who has *re-tweeted* frequently in the past is likely to *re-tweet* in the future.

- Most users are only interested in *tweets* under certain event categories. Most followers are users who have similar interests.

- Users' interests and preferences are assumed to be stable.

In this section, we present the details of our approaches to predict the volume of *re-tweets* and possible views for given original short messages (*tweets*). We study the problem of modelling users' behaviours by focusing on *re-tweeting* activities in order to understand the users' participation for spreading information in social networks. The model can be used in many applications such as marketing and recommendation. Towards this, we propose four applicable different approaches. The first method is a naïve approach that discovers a regression function based on the popularity of messages and network connectivity. The second approach is to build a classifier that learns a classification model based on the user's preferences in different categories of topics. The third approach focuses on a network simulation that leverages a *Monte Carlo* method to simulate *re-tweeting* paths starting from a root message. The fourth approach uses collaborative filtering to build a recommendation model. The results of these four methods are compared in terms of their effectiveness and efficiency.

Approach 1: Regression based on Popularity and Connectivity

We develop a model to predict *re-tweet* activities based on event popularity and user connectivity by using a naïve approach. The intuition is that a *tweet* is more likely to be *re-tweeted* if it is about a popular event and its author is highly connected with others. The prediction will be the estimation of the probabilities of these two parameters in the space (connectivity of the user and category popularity).

The prediction of *re-tweets*:

With regard to the prediction of *re-tweets* for any given *tweet* and the *user ID*, we find its connectivity of the user and its popularity of the event. To compute the connectivity of a user, we design a function $C(uid)$ to find how many *re-tweets* a uid (*user ID*) may have based on the number of followers the user has. We divide users into 10 groups according to the number of followers (shown in Table 2.1) and randomly pick 10 percent of users from each group to calculate the average number of *re-tweets* as the group statistics.

To compute event category popularity, we design a function $P(uid, category)$ to predict how the

TABLE 2.1: 10 groups of users according to the number of followers.

Group	Number of followers	Number of users	%
1	< 10	2,573,915	91.30
2	10-19	32,396	1.15
3	20-29	9,637	0.34
4	30-49	8,609	0.31
5	50-99	14,235	0.50
6	100-499	81,018	2.87
7	500-999	43,206	1.53
8	1,000-4,999	46,338	1.64
9	5,000-9,999	7,946	0.28
10	$\geq$ 10,000	2,024	0.07

event category popularity influences a *tweet* being *re-tweeted*. For a given user, the average number of *re-tweets* is computed for each category in three levels within 30 days. Events are manually grouped into categories as described in Section 1.2.2. If a given user has never posted any *tweet* belonging to the event category, we use the average number of *re-tweets* of the "*No Category*" instead. The formula for *re-tweet* prediction is shown as Eq. 2.1.

$$\#RTs = s(\alpha C(uid) + (1 - \alpha)P(uid, category)) \tag{2.1}$$

where s is a scaling factor, as Functions C and P only consider up to three levels and α is a scaling of weights between Function C and P. Parameter α has been learned from statistics computed from the micro-blog dataset. We found that the ratio of *re-tweeting* by followers to non-followers is 3,352,996:261,811,184 (0.024:0.976). Therefore, we set α as 0.024. The scaling factor s has been derived from *re-tweet* records in the training set. We randomly pick 1,000 *tweets* and calculate the average of s. This process is repeated 10 times and the overall average s is 19.95. Our equation is shown below:

$$\#RTs = 19.95(0.024C(uid) + 0.976P(uid, category)) \tag{2.2}$$

The prediction of possible views:

In order to predict the number of possible views of a given original *tweet*, we aim to count the number of followers of users who will *re-tweet* the original *tweet*. According to statistics, the percentage of *re-tweets* at the first three levels are 61.73%, 28.50%, and 9.77% respectively. So, we assume

that the number of *re-tweets* ($\#RTs$) computed from Eq. 2.2 is distributed accordingly. For example, if the number of *re-tweets* is 1,000 times then the number of *re-tweets* in the first level is 617 times and the number of *re-tweets* in the second and third levels are 285 and 98 times respectively. For each level, we randomly select l users who have a history of *re-tweeting* in the same event category of the original *tweet*. If the number of users in the category is less than l, we randomly select users from "*No category*". In this case, it is possible that the randomly picked users have no followers. Therefore, we repeat the process ten times and compute the average as the resulting prediction. If the number of followers of a given user who posts the original *tweet* is zero, the equation is

$$PViews = rtUsers + FollowerRTs \tag{2.3}$$

Otherwise, the equation is

$$PViews = FollowerU + 0.976(rtUsers) + FollowerRTs \tag{2.4}$$

where $FollowerU$ is the number of followers for a given user who posts the original *tweet*, $rtUsers$ is the number of *re-tweeters*, and $FollowerRTs$ is sum of the number of followers of *re-tweeters*.

Approach 2: Classification based on User Preferences

As people's interests may differ, their interests in the types of *tweets* will also differ. We refer to this phenomenon as "user preference". User preferences are used to train a classifier to predict the possible number of *re-tweets* and the possible number of views in 30 days for a given original *tweet*. We firstly pre-process the *re-tweeting* dataset to assign users to certain event categories according to their *re-tweeting* activities. One of our event categories is called "*No Category*" to indicate that the interests of the user are unknown. In our method, the process includes three steps as follows:

Interestingness Computing:

Given an original *tweet*, we need to compute how likely a user will *re-tweet* the original *tweet* in the category. The candidate users are extracted from *re-tweet* history in the form of "*who-retweet-who*". We use $P(r, u, c)$ to denote the interestingness of the candidate *re-tweet* user r to original user

u on category c. The function is defined as Eq. 2.5.

$$P(r,u,c) = \sum RT(r,u,c) / \sum T(u,c) \tag{2.5}$$

where $RT(r,u,c)$ returns the number of *re-tweets* by user r from user u on category c; $T(u,c)$ returns the total number of u's *tweets* on category c.

Algorithm 1: $PredictRetweet()$

Input: u:user id, c:category, curr:current retweet number, level:current level, maxLevel:max followers level, lamb:threshold value
Output: rtnum:total retweet number

```
1  rtnum = 0;
2  if level < maxLevel then
3      //Get all the followers of current user u
4      list= GetAllFollowers(u);
5      for i=0 to list.count do
6          f = list[i];
7          p = 0;
8          //Using P(f,u,c) to get the user history interest score
9          //Then compare with the threshold value lamb
10         if p = P(f,u,c) ≥ lamb then
11             p = 1;
12             rtnum += p;
13             //Compute the next level
14             rtnum = PredicRetweet(f,c,rtnum,level+1,maxLevel,lamb);
15 return rtnum;
```

Classifier Training:

We build a classifier using user interestingness scores. During training, the classifier categorized every candidate user to "*re-tweet*" or "*no-retweet*", labelled as 1 or 0 respectively. For a candidate user, if the user has a high interestingness score on a category, the candidate user is likely to *re-tweet* the original user's *tweet* in the future. We use a threshold value λ to build the classifier, where $\lambda \in$ [0,1]. The classifier $Q(r,u,c)$ is defined as:

$$Q(r,u,c) = \begin{cases} 1, \ if \ P(r,u,c) \geq \lambda \\ 0, \ if \ P(r,u,c) < \lambda \end{cases} \tag{2.6}$$

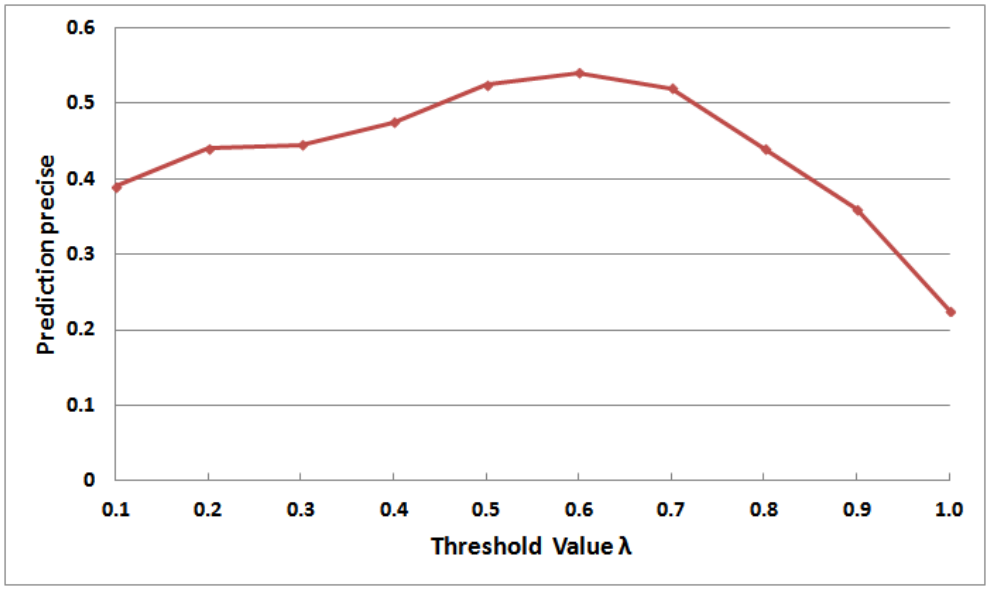


FIGURE 2.1: Prediction accuracy with different lambda.

In order to find the most suitable value for threshold λ, we carry out predictions on *tweets* from the training dataset with different λ values. The results are shown in Fig. 2.1. Our tests show that when $\lambda = 0.6$ it renders the best performance.

Prediction:

We use the classifier described above (Eq. 2.6) and the function given in Eq. 2.5 to predict the possible *re-tweets* and views for each given *tweet*. The details are explained as follows:

1. Given an original *tweet*, get its current *re-tweets* and *re-tweeters* before a given timestamp.

2. For each current *re-tweeters*, get its followers from the followship dataset as the next-level *re-tweet* candidates.

3. For each candidate *re-tweeter* r, compute its interestingness score and use the classifier to classify if r will *re-tweet* the original *tweet*.

4. Accumulate the predicted number of *re-tweets*.

5. Repeat 1-4 until no more *re-tweets*.

6. Return the total number of *re-tweets*.

Algorithm 2: $PredictView()$

Input: u:user id,c:category, prepos:predicted possibility, curr:current retweet number,level:current level,maxLevel:max followers level,lamb:threshold value
Output: view:total view number

```
1  view = 0;
2  if level < maxLevel then
3      list= GetAllFollowers(uid);
4      //Compute the current level of retweet
5      view += (prepos * list.count);
6      for i=0 to list.count do
7          f = list[i];
8          if p = P(f, u, c) ≥ lamb then
9              //Compute the next level of retweet
10             view = PredictView(f, c, rtnum, level + 1, maxLevel, lamb);
11 return view;
```

The number of possible views is computed based on the number of followers at every level. For a given *tweet* at the current level, if the current candidate *re-tweeter* r's *re-tweet* possibility is p and the number of r's followers is n, the current number of possible views is $p * n$.

Prediction Algorithm:

Here we introduce the prediction algorithm. According to the prediction method we introduced above, we design the prediction algorithms: $PredictRetweet()$ and $PredictView()$ algorithms. The $PredictRetweet()$ algorithm is to predict the possible *re-tweeters* for a given *tweet* in thirty days, the algorithm is shown as Algorithm 1.

In $PredictRetweet()$ algorithm, $GetAllFollowers(u)$ is a function to obtain all the followers of user u from the followship dataset. The $PredictView()$ algorithm predicts the number of possible views for a given *tweet* in thirty days, the algorithm is shown as Algorithm 2.

Approach 3: Network Simulation Approach based on *Re-tweeting* Behaviours

The network simulation approach attempts to simulate the *re-tweeting* propagation starting from a root user who has posted an original *tweet*. Every user in the network is viewed as a node in a graph of users simulating the *re-tweeting* behaviours. This probabilistic approach is equivalent to a spanning tree from the root with probabilities predicting at a current user node, 1) if the *tweet* will be *re-tweeted*,

Algorithm 3: $PredictRetweetBySimulation()$

Input: rtMid:ID of an original tweet,rtUid:ID of the author of rtMid, event:event associated with rtMid RtNet:Sina Weibo re-tweet data set, FollowNet: Sina Weibo follower data set, alpha: the probability that a FollowNet is used

Output: RT:the predicted number of total re-tweets, PV:the predicted number of total possible views

```
1  eventCategory =: getCategory(event)
2  numFollowers =: getNumFollowers(rtUid, FollowNet)
3  Q =: {rtUid,0,numFollowers}, PV=0, RT=0
4  postGroupDistribution =: getDistribution(RtNet)
5  while Q ≠ ∅ do
6      ⟨curUid, curLevel, numFollowers⟩=: Q.dequeue()
7      RT+=1, PV+=numFollowers
8      trend =: getTrend(curUid, eventCategory, RtNet)
9      ϵ =: getEpsilon(trend, curLevel)
10     randomGenerator.initiate()
11     if randomGenerator.next() <= ϵ then
12         numToPick =: getNumToPick(randomGenerator, trend)
13         selected:=∅ for i=0 to numToPick do
14             if randomGenerator.next() <= α then
15                 followers =: getFollowers(curUid)
16                 distribution =: normalizedFollowerPostedRT(followers)
17                 selected.add(randomPick(distribution, followers, randomGenerator))
18             else
19                 selected.add(
20                 randomPick(postGroupDistribution, randomGenerator))
21         foreach uid u of selected do
22             Q.enqueue(⟨uid, curLevel + 1, uidNumFollowers⟩)
23 return ⟨RT, PV⟩;
```

2) how many *re-tweets* will be received by the user, and 3) who will be the further *re-tweeters*. To address these three aspects, a model is built based on the following three factors.

First-level *Re-tweet*:

A first-level *re-tweet* is a *re-tweet* made directly from an original *tweet*. We found that the number of first-level *re-tweets* accounts for more than 50% of the total *re-tweets* that an original *tweet* had received. For this reason, the average first-level *re-tweets* of the author of an original *tweet* is used to predict the number of times her *tweet* will be *re-tweeted* directly.

First-level User Group and Event Trend:

Given an event, users in the simulated network are partitioned into groups at an interval of 10 bases, according to their average first-level *re-tweets* in the past. A trend of an event within a group is the average *re-tweets* on the event from levels 1 to 20. The ratio between two subsequent levels is used as the probability that a current *tweet* will be *re-tweeted* at the next level. The average number for the next level is also used as the upper bound for randomly selecting the number of *re-tweets* to be received at the next level based on a probability distribution. This trend model is built based on the *re-tweet* history of all events in the same event category.

***Re-tweet* User Group**:

We found that only 2% of *re-tweets* were made by followers and thus the probability of using the followship network is 2%. If a *re-tweet* is made by a follower, a follower is randomly selected according to a probability distribution based on the total number of *re-tweets* posted by followers. Otherwise, a user is randomly selected based on the *re-tweet* group probability distribution. Users are categorized into *re-tweet* groups based on the total number of *re-tweets* they posted in the past (shown in Table 2.1). The average number of *re-tweets* posted by the group users is used to build a probability distribution over all groups. The implementation of this approach followed a *Monte Carlo* method and our algorithm is shown as Algorithm 3.

Approach 4: Collaborative Filtering based on *Tweet* Similarity

Collaborative filtering has been widely applied for the prediction of interest ratings for a given item by using existing user profiles [121, 99]. In general, collaborative filtering algorithms are divided into two basic types: model-based approaches and memory-based ones. Among most of the former approaches, a number of parameters need to be estimated or tuned to complete the prediction which is not applicable in practice. In the latter approaches, similar user and item profiles are memorized by certain types of sort to predict ratings, which requires significantly fewer parameters than the model-based methods. In our approach, we apply item-based collaborative filtering to predict the numbers of *re-tweets* and possible views. We process all messages related to events that are officially classified, as different items which have been rated or will be rated by users. Each *tweet* is regarded as an item and is associated with an event category.

Data Structure:

Given M items and K users, the user profiles can be represented by a $K \times M$ matrix called the user-item matrix X. All row vectors, u_k, $k = 1, \ldots, K$, and column vectors, i_m, $m = 1, \ldots, M$, represent one particular user's ratings for all items and all users' ratings for one item, respectively. In item-based collaborative filtering, the prediction of a test item by a test user, relies on ratings on similar items for a user. In our case, all messages expected to be predicted are completely unknown items for existing users. Specifically, a prediction on a test item m by a test user k is represented by $\hat{x}_{k,m}$. Accordingly, the prediction can be formulated as follows:

$$\hat{x}_{k,m} = PI_{k,m}(x_{k,b}), x_{k,b} \in \mathcal{S}_i(i_m), x_{k,b} \neq \emptyset \tag{2.7}$$

where $PI(\cdot)$ in Eq. 2.7 is a prediction function based on item similarity. $\mathcal{S}_i(i_m)$ is a set of similar items to item i_m for user u_k .

The prediction function can be constructed in different ways. We have compared a factor graph-based approach [120] with our approach that is designed to count the average number of *re-tweets* per event. We find that our approach performs much faster with similar rating values because *tweets* in our approach are featured in terms of 12 event categories.

Item Similarity Measurement:

In our case, each individual message *tweeted* by a user is viewed as an item and the value of recurrences is regarded as a rating value discovered by the out counting function. The more *re-tweets* an original *tweet* receives, the higher rating the item receives. Meanwhile, each message can be classified as an instance of an event. In other words, one event can raise various discussion instances. For example, the event *"House prices"* can be associated with a large number of messages with *ids*: *"014135972192"*, *"014135642802"* and etc.

Though *tweeters re-tweet* these messages from different sources, they should be still classified as the same type. In light of this, comparisons of similarity between messages are equal to comparisons of similarity between events. Officially, 46 events are provided together with their corresponding keyword list. Unfortunately, it is hard to compare their semantic concepts because some events are simply named by particular personal names, e.g. *"Qian Yunhui", "Yao Jiaxin Murder"*, or comprised

of some particular place names, e.g. *"Fuzhou Bombing", "Anshun incident"*. Moreover, their relevant keyword lists are much more arbitrary. To solve this problem, we manually label all events with some attributes and calculate the similarity between two events by using these attributes. We assign "0" or "1" to each attribute field for each event and then we obtain 12 dimensional binary feature vectors for each event for 12 predefined event categories. For two arbitrary *tweets* Tw_A and Tw_B, we apply the cosine similarity metric upon their dimensional event feature vectors as Eq. 2.8.

Similarity between events at the semantic level can now be measured. For example, the event *"death of Steve Jos"* and *"the death of Osama Bin Laden"* are neighbourhood. In this way, the similarity comparison of different event messages is replaced by the comparison of their corresponding events. Moreover, for all messages within one event, the message features are used to distinguish each event message.

$$Similarity(Tw_A, Tw_B) = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \tag{2.8}$$

We assume that *tweets* posted within a time frame contain similar concepts or close topics. When an event happens, a variety of *tweets* would be posted. Over time, newer discussions in *tweets*, about the same topic, are likely to commence. Moreover, the average duration starting from the time of posting to the time when it is *re-tweeted* can reflect the popularity of this *tweet*. We term this as the average response time and use it as a feature in our method for prediction. The numbers of predictions are formulated as follows:

$$\begin{aligned} \#retweet_p = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} r \times x_{i,j}, x_{i,j} \neq 0, \\ x_{i,j} \in \mathcal{S}_i(retweet_p) \end{aligned} \tag{2.9}$$

$$\begin{aligned} \#viewer_p = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} f(x_{i,j}), x_{i,j} \neq 0, \\ x_{i,j} \in \mathcal{S}_i(retweet_p) \end{aligned} \tag{2.10}$$

Note that $\|\mathcal{S}_i(\cdot)\| = N, N \ll M$, r is a parameter that controls the ratio of popularity and $f(\cdot)$ is a

function which returns the number of followers of a user. The algorithm is shown as Algorithm 4.

Algorithm 4: $PredictRetweetandViewerNumbers()$

Input: mid:message id; $eventDistanceMatrix$: Distance Matrix for all events; K: K nearest neighbour events of mid's event; $retweetList.length$: all re-tweet records in the training set; N: N nearest neighbour messages of mid; UI:User-Item Matrix

Output: num_r:predicted number of re-tweets, num_v:predicted number of viewers

$eventId = GetMessageEvent(mid)$;
$simEventList = GetSimilarEvents(eventDistanceMatrix, eventId, K)$;
for $i = 1$ **to** $retweetList.length$ **do**
    $eventId = GetMessageEvent(retweet_i)$;
    if $eventId \in simEventList$ **then**
        $s = CompareSimilarity(retweet_i, mid)$;
        $sim\_message\_list.insert(< s, retweet_i >)$;
$sim\_message\_list.sort("ascend")$;
if $sim\_message\_list.length < N$ **then**
    $N = sim\_message\_list.length$;
$sim\_message\_list = sim\_message\_list(1 : N)$;
for $i = 1$ **to** $sim\_message\_list.length$ **do**
    $sum\_rating_i = 0$;
    $sum\_viewer_i = 0$;
    for $j = 1$ **to** $userList.length$ **do**
        if $GetRating(UI(user_j, item_i)) \neq 0$ **then**
            $sum\_ratings_i += GetRating(UI(user_j, item_i))$;
            $sum\_viewer_i += GetFollowerNum(user_j)$;
    $ratio_i = \frac{GetPopularity(mid)}{GetPopularity(retweet_i)}$;
    $sum\_rating_i \times = ratio_i$;
$num_r = \frac{1}{N} \times \sum_{i=1}^{N} sum\_rating_i$;
$num_v = \frac{1}{N} \times \sum_{i=1}^{N} sum\_viewer_i$;
return num_r,num_v;

Experiments and Evaluations

In the training dataset, we removed all 33 *tweets* belonging to the test events of the *WISE 2012* Challenge. For the classification approach (i.e., Approach 2), two datasets are used. For training, we extract 970,125 original *tweets* and 5,690,837 *re-tweet* records by 330,386 *re-tweet* users. The computations of four methods are all conducted on the PC computers with Core(TM) i7 vPro 2.93

GHz Intel processors and 4 GB of RAM. The average predicting times for each *tweet* of the four proposed approaches are shown in Table 2.2.

TABLE 2.2: The average prediction time (second).

Approach	1	2	3	4
Time	4	70	172	15

For evaluation purposes, we randomly select 50 messages from the testing set 10 times and compute the average. Our evaluation formulae are shown below:

$$Prediction\_error = |Actual - Predict|/Actual$$

$$Accuracy = Num\_CorrectMsgs/50 \tag{2.11}$$

$$Average\_Accuracy = \sum_{i=1}^{10} Accuracy_i/10 \tag{2.12}$$

We compute the $Prediction\_error$ value which is the difference between the actual value and its predictor for every message. Then, we define the message which has $Prediction\_error \leq Threshold$ as the correctly predicted message and count it as $Num\_CorrectMsgs$. $Threshold$ is an acceptable prediction error value. $Accuracy$ is the percentage of messages considered as correctly predicted values for each round, and $Average\_Accuracy$ is the average number of $Accuracy$ in 10 rounds. The performance of the four approaches are shown in Table 2.3.

TABLE 2.3: Prediction accuracy on different error rates of four approaches

Acceptable error threshold	**Number of re-tweets**				**Number of Possible views**			
	1	2	3	4	1	2	3	4
0.05	0.02	**0.14**	0.05	0.09	0.05	**0.06**	0.03	0.04
0.10	0.05	**0.30**	0.09	0.19	0.11	**0.14**	0.03	0.09
0.20	0.12	**0.52**	0.14	0.32	0.20	**0.20**	0.05	0.11
0.30	0.22	**0.60**	0.22	0.47	0.26	**0.29**	0.09	0.19
0.40	0.23	**0.70**	0.31	0.62	0.35	**0.39**	0.15	0.24

In summary, we find that each approach has its own advantages and disadvantages. The predictions of the first approach (i.e., regression) are based on user connectivity and event popularity of the previous similar event category. The limitation of this method is that the predicted results can be the

same when given the same user id and the same event but different message ids. The second approach (i.e., classification) makes predictions based on users' *re-tweet* preferences in different *tweet* categories. The prediction accuracy is dependent on the partitioning of categories and the stability of user preferences. Predictions made by the third approach (i.e., network simulation) highly rely on the average first level re-tweets of the authors whose original *tweet* is to be predicted. Hence, the performance is highly dependent on the authors' history. The fourth approach (i.e., collaborative filtering) obtains the prediction results of messages by considering contributions from the top N similar messages which need less tunes of parameters. However, both results of *re-tweet* number and possible views are heavily dependent on the item similarity metric. Moreover, the predicted number of possible views partially applies the hierarchical structure of social networks leading to an ignorance of *tweet-retweet-network* structure.

2.2 Event Detection in Social Networks

2.2.1 Event Detection in Social Streams

Online new event detection and tracking which is part of topic detection and tracking (TDT) was first studied by Allan et al. [3]. Mining online news for events was a hot topic in information retrieval during the last decade [118, 49, 59, 24]. Event detection on micro-blogs as a challenging research topic has been increasingly reported recently [96, 91, 8, 114, 57, 78, 2, 56], while other research considers event identification in other social media data such as *Flickr*[1] - an online photo management and video hosting website, and web services suite [85, 14, 7, 58].

Sayyadi et al. in [96] develop a new event detection algorithm by using a keyword graph and community detection algorithm to discover events from social streams. A *KeyGraph* is built by first extracting a set of keywords. Keywords with low document frequency are filtered. Each remaining keyword is represented as a node in the *KeyGraph*. An edge is created between two nodes if the two keywords co-occur in the same document. An edge is removed if it does not satisfy the two conditions first, its nodes co-occur below some minimum threshold and second, if the conditional probability of

[1] https://www.flickr.com/about

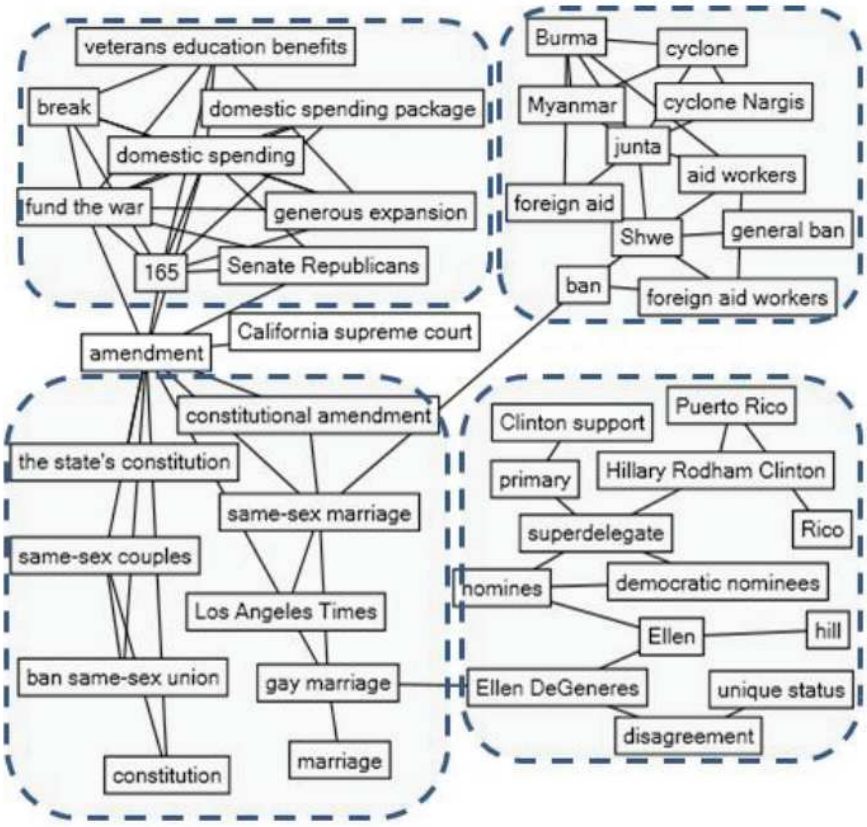


FIGURE 2.2: A sample of the *KeyGraph* and communities of keywords [96].

the occurrence and similarity between the two terms are smaller than the defined thresholds. Communities of keywords are detected by removing the edges with a high betweenness centrality score. For each community, cosine similarity is used to discover document clusters for key documents. However, the number of detected events depends on the threshold parameters and there was no evaluation conducted. A sample of the *KeyGraph* and communities of keywords is shown in Figure 2.2.

Sakaki et al. in [91] present a real-time event detection approach by using *Twitter* messages associated with time and geographic location information to detect event occurrences. This work is concerned with quickly detecting specific types of events (i.e., earthquakes, typhoons and traffic jams) in order to issue a timely warning for the areas that were about to be affected by these disasters. The authors manually define a set of keywords relevant to the types of events they want to detect such as {"earthquake" and "shaking"} in the earthquake situation and {"typhoon"} in the typhoon situation. For each message, the Support Vector Machine (SVM) is used to classify whether it is about an event or not. Three groups of features for each message are used including statistical features (i.e., the

number of words and the position of the query word within the message), keyword features (i.e., the words in a message), and word context features (i.e., the words before and after the query word). Each *Twitter* user is regarded as a sensor for detecting a target event. Each message is associated with a time and location (i.e., a set of latitude and longitude). The event is identified if there are enough messages thate were classified as being about an event occurring in a short time period. The correspondence between event detection from *Twitter* and object detection in a ubiquitous environment is presented in Figure 2.3. However, this approach needs to manually define a set of keywords for each event. Also, it requires labelled data to train classifiers for every event type.

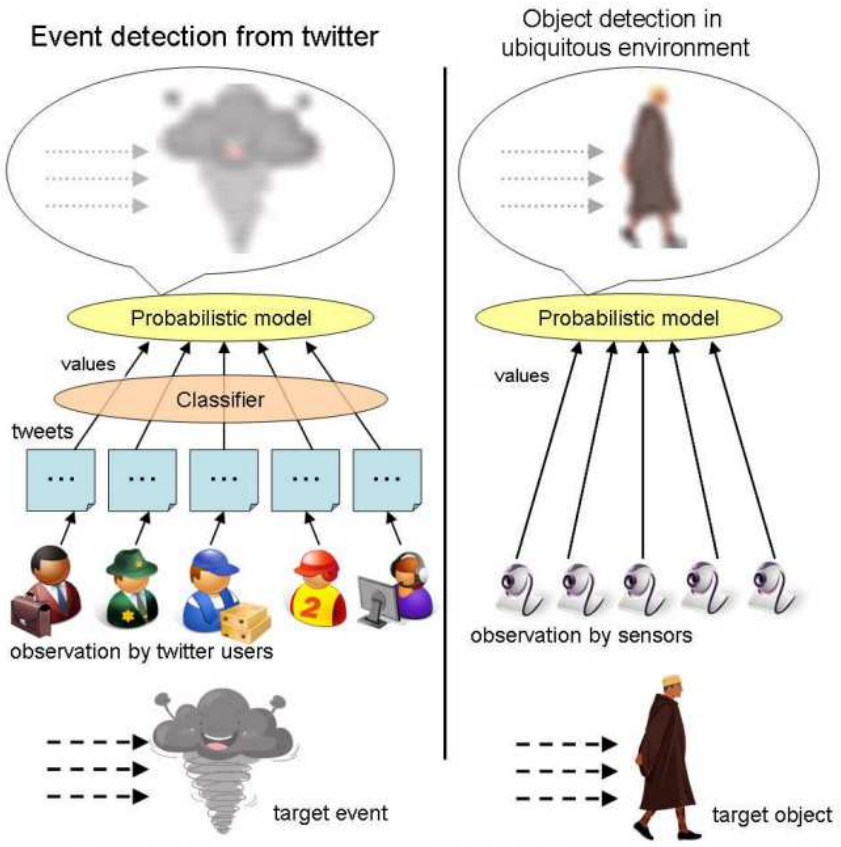


FIGURE 2.3: The correspondence between sensory data detection and *Twitter* processing in [91].

Becker et al. in [8] apply an online clustering algorithm and classifier to distinguish between messages about real-world events and non-event messages. The authors use an online clustering algorithm because it does not require a priori knowledge of the number of clusters. Each message is presented as a *tf-idf* weight vector. The centroid is used to represent each cluster. The cosine similarity metric is used as the clustering similarity function. The authors consider temporal, social, topical and

Twitter-centric features to identify event clusters. The classifier is used in a post-processing step for classifying the cluster whether it is about a real-world event or not. The authors use a set of features derived from all the messages in the cluster such as the number of *re-tweets*, or the number of messages containing the most popular *hashtag*. However, this approach relies on having labelled data to train a classifier and it is not clear if retraining the classifier is needed.

An approach, named *EDCoW* is proposed by Weng et al in [114]. The authors use wavelet transformation and auto correlation to measure the bursty energy of each term. DFIDF (Document Frequency-Inverse Document Frequency) is used to measure a term's importance in a certain time period. Then, cross-correlation is computed for those terms that show significant change over time. Modularity-based graph partitioning is used to detect the events, each of which contains a set of terms with high cross-correlation. However, wavelet transformation and auto correlation for each term of the *Twitter* messages would require a huge amount of computation. Moreover, the performance is sensitive to the parameter setting.

A *Twitter* based Events Detection and Analysis System (*TEDAS*) is proposed by Li et al. in [57]. The focus of this approach is on Crime and Disaster related Events (CDE). It consists of three functions including (1) new events detection, (2) events ranking and (3) temporal and spatial patterns generation for events. A classifier is used to determine whether a message is related to a CDE, while the online processing supports the CDE's detection, answers users' analytical queries and generates visual results. *Twitter*-specific and CDE-specific features are used to train the classifier. The *Twitter*-specific features include a short URL, hash tag and an "@" sign. CDE-specific features include any time or location mentioned in the message, and any frequently mentioned number in a message. However, this approach relies on having labelled data to train a classifier for a particular type of event.

Ozdikis et al. in [78] propose an event detection method in *Twitter* based on the clustering of *hashtags*, the "#" symbol used to mark keywords or topics in *Twitter*, and apply a semantic expansion to message vectors. For each *hashtag* the most similar three *hashtags* are extracted by using cosine similarity. A *tweet* vector with a single *hashtag* is expanded with three similar *hashtags* and then used in the clustering process. However, using messages with a single *hashtag* can suffer from the problem of ignoring all messages that do not contain a *hashtag*. Also they do not implement any credibility filter in order to decide whether a *tweet* is about an event or not.

Agarwal et al. in [2] apply a graph clustering algorithm for real-time event detection. Their approach relies on discovering highly dense clusters in graphs where nodes are keywords, and edges correspond to a user using both words in their tweets. However, it can only detect sufficiently bursty events. This approach ignores non-bursty events and does not measure the amount of meaningless or irrelevant events that are detected. Moreover, it is very hard to interpret the results when the evaluation is based on words and not documents.

Li et al. present *Twevent* in [56]. It is a state-of-the-art system detecting events from the *tweet* stream. The authors use the notion of *tweet* segments instead of unigram to detect and describe events. Given *Twitter* messages, *Twevent* firstly segments each individual message into a sequence of consecutive phrases by using *Microsoft Web N-Gram*. Then bursty segments are identified by modelling the frequency of a segment. User frequency of the *tweet* segments is used to identify the event-related bursty segments. Then, a clustering algorithm is applied to group event-related segments as candidate events. *Wikipedia* is utilized to approximately evaluate important and unusual aspects of a candidate event. The system architecture of *Twevent* is shown in Figure 2.4. As a result, the events detected with *Twevent* are heavily influenced by *Microsoft Web N-Gram* and *Wikipedia*, which could potentially distort the perception of events by *Twitter* users and also give less importance to recent events that are not yet reported on *Wikipedia*.

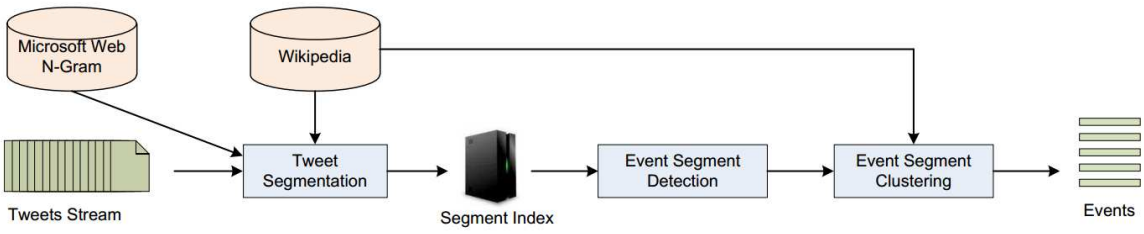


FIGURE 2.4: Segment-based event detection system architecture [56].

2.2.2 Emerging Topic/Event Detection

A significant amount of research has previously been conducted on emerging topic detection [67, 12, 29, 4, 18]. The work by Budak et al. [11] present the network topology of friendship information to identify the trend of the topic but they do not include topic extraction. Sarma et al. [95] propose

an approach to detecting temporal relationships between entities. The authors extract co-peaking entities from the message stream, which are grouped according to co-occurrence. Matheoudakis et al. present *TwitterMonitor* in [67], a system that performs trend detection over the *Twitter* stream. The system detects emerging topics in real-time. First, the system identifies bursty keywords. Then, bursty keywords is grouped into trends based on their co-occurrences. Finally, additional information is extracted from the messages that belong to the trend. This work focuses on trend detection rather than event detection.

Goorha et al. in [29] present a system for the automatic identification of emerging topics associated with a products of interest. Cui et al. in [18] propose an approach for discovering breaking events with popular hashtags in *Twitter*. The authors categorize hashtags to discover vocabulary associated with news, filtering our memes, idioms, and advertisements.

Cataldi et al. discuss emerging topic detection on *Twitter* [12]. Their work uses aging theory to model the life cycle of each term. The importance of *Twitter* users is studied, which represents an important weighting of contents. An authority score is computed for each user using the number of followers and the number of followees. Emerging terms are identified within a given time interval. Terms that frequently correlate with emerging terms are extracted and are reported together as emerging topics. A topic graph is constructed in the form of a directed, node-labelled graph. Strongly Connected Components (SCCs) are extracted to represent emerging topics. However, the system needs to compute user-authority values for weighting terms but it is difficult to collect a complete user-network.

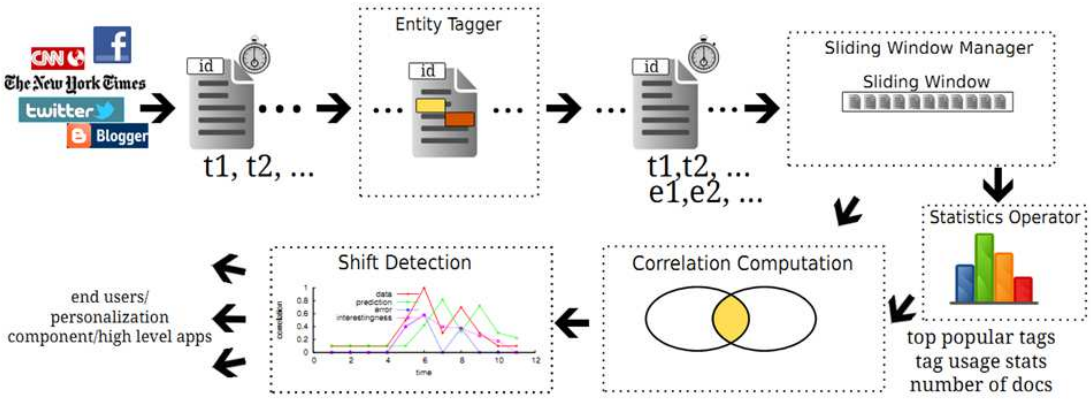


FIGURE 2.5: A framework of *EnBlogue* [4].

EnBlogue is presented by Alvanaki et al. in [4]. This approach consists of three steps: seed-tag selection, correlation tracking and shift detection. This approach tracks the correlation of tag pairs which contain at least one seed tag. The seed tags are selected from the current sliding window based on term frequency. To measure tag correlations, the Jaccard coefficient is used to measure local and global importance. Exponential smoothing, which uses a weighted moving average of past data, is used for the forecast. A topic is emergent if its real correlation is larger than the predicted value. Emerging events are identified as the top k highest shift scores of tag pairs. A post-processing is performed to group tag pairs that refer to the same event. Two tag pairs are grouped together if they co-exist in 80 percent of the messages. However, the performance is highly sensitive to the selected seed tags. As this approach is based on clustering pairs of terms, this makes it very difficult to interpret the results. The framework of their approach is shown in Figure 2.5

2.2.3 Location-based Event Detection

The geographical scope of social networks content has been studied in the last decade. Serdyukov et al. propose methods for automatically placing photos uploaded in *Flickr* on the world map [97]. They use the terms people employ to describe images to analyse a particular location while Cheng et al. propose a probabilistic framework for estimating a *Twitter* user's city-level location based on the *Twitter* messages [16]. They create a classifier which identifies tweet words with local geo scope and return the top k cities for each user. However, their approach needs to gather a sufficient amount of training data and they are not considering the time period.

TwitterStand is proposed by Sankaranarayanan et al. in [94]. Two thousand hand-picked users of *Twitter* are used as seeders who are known to publish news. The on-line clustering method is used to group the messages into the news topics. User location and content location are used to locate geographic content from each news topic. To handle noise, a pre-processing step is added which classifies each message as being about news or not, by using a Naïve Bayes classifier. However, this approach relies on having labelled data to train classifiers and the results also rely on having hand-picked users. Moreover, there is no evaluation conducted. The system architecture is shown in Figure 2.6.

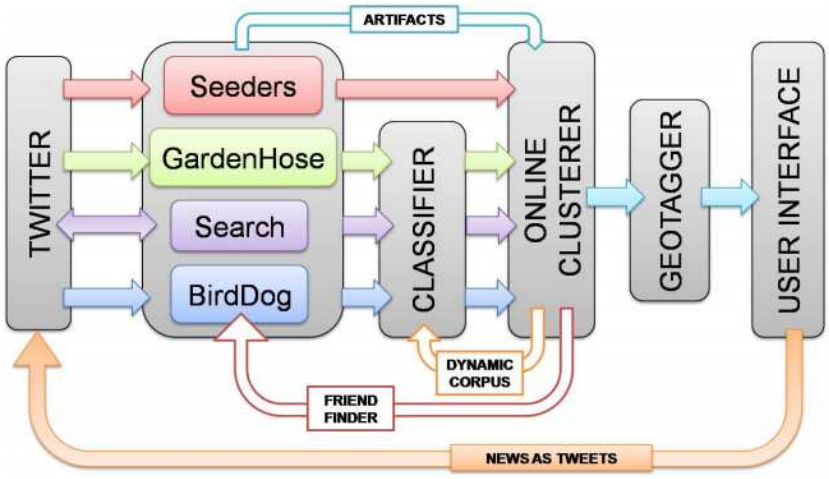


FIGURE 2.6: A system architecture of *TwitterStand* [94].

Furthermore, there are several studies that are focused on identifying events of a particular type such as forest fires, earthquakes, and swine flu. Longueville et al. [63] describe the use of *Twitter* during a forest fire. They identify different types of *Twitter* users: those related to mass media outlets, those acting as aggregators of information, and normal citizens. Sakaki et al. [91] investigate the real-time interaction of events in *Twitter*, such as earthquake and propose an algorithm to monitor tweets and to detect a target event. They also apply Kalman filtering and particle filtering for estimating the centres of earthquakes and the trajectories of typhoons by using latitude and longitude content of *tweet* messages and the registered location of a user. In addition, Singh et al. [98] propose a new way of organizing spatio-temporal micro-blog data into social images. They demonstrate the use of simple user-defined bag-of-word models to capture relevant user interest for any time-window at a given geo-location.

TwitterReporter is proposed in [71] by Meyer et al. These authors present methods to collect data, identify breaking news topics, and display results in a geo-temporal visualization. The topics are grouped into three categories: natural events, man-made events, and other uncategorized events. For example, a natural event includes “tornado”, “earthquake” and “hurricane”. The Document Frequency (DF) is computed for a given term within the entire batch of messages. An approach for event detection by mining spatio-temporal information on micro-blogs is proposed in [51] by Lee et al. They present several algorithms to effectively detect and group emerging topics by making

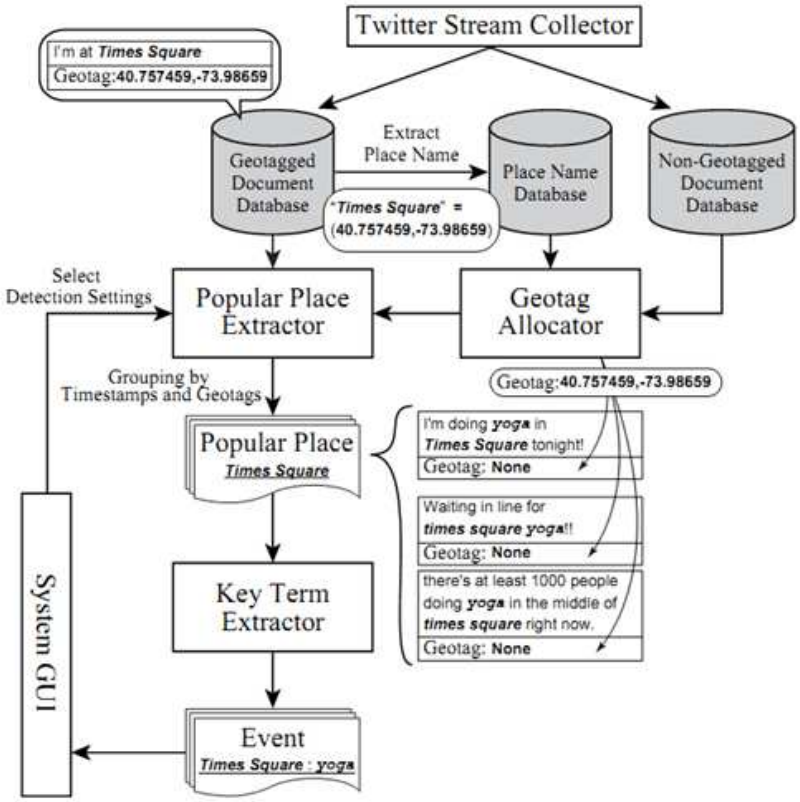


FIGURE 2.7: A system architecture of *Jasmine* [112].

use of real-time messages and geo-location data provided by social network services. Messages are clustered based on incremental *DBSCAN*. The location of each cluster or event is estimated by using users' time zones.

Watanabe et al. propose a real-time local-event detection system named *Jasmine* in [112]. The system architecture is shown in Figure 2.7. Local events are detected by using *geo-tagged* information (latitude and longitude) from *Twitter* data. The place name is extracted from check-in messages such as *"I'm at Time Square"*. The researchers search for *non-geo-tagged* messages which contain a distinct place name and allocate a *geo-tagged* information to the location. However, the results rely on the number of *geo-tagged* data they had. It is also difficult to determine the location when more than one location has the same place name, such as the restaurant chains (e.g., *"I'm at McDonald's"*) or the supermarket chains (e.g., *"I'm at Coles Supermarket"*).

2.2.4 Sub-Event Detection from Social Networks

Micro-blog like *Twitter* has been attracting growing attention from researchers in Data Mining and Information Retrieval. Extensive research has been done on social networks in event detection [2, 4, 8, 12, 56, 67, 114]. However, there exists few researches on engineering the search and retrieval of relevant information from social network data [1, 66, 83].

Abel et al. in [1] introduce *Twitcident*, a framework and web-based system for filtering, searching and analysing information about real-world incidents or crises. Given an incident, the system automatically collects and filters relevant information from *Twitter*. When a new message is posted, it searches for related *tweets* which are semantically extended in order to allow for effective filtering. Users may also make use of a faceted search interface to delve deeper into these *tweets*. However, this work focuses on how to enrich the semantics of *Twitter* messages to improve the incident profiling and filtering rather than to detect sub-events and users' opinions of each event.

Pohl et al. in [83] propose automatic sub-event detection in emergency management using social media. Researchers perform the sub-event detection using a Self-Organizing Map clustering approach on *Flickr* and *YouTube* data (i.e., photo and video datasets) which is different from our work that only focuses on text messages.

A research which is similar to our work is presented by Marcus et al. in [66]. A system for visualizing and summarizing events on *Twitter* in real-time, namely *TwitInfo*, is proposed. The system detects sub-events and provides an aggregate view of user sentiment. Sub-events are extracted by identifying temporal peaks in message frequency and by using the weighted moving average and variance to detect an outlier as a sub-event. The *Naïve Bayes* classifier is used to analyse the sentiment of messages into positive and negative via *unigram* features. Training datasets are generated for the positive and negative classes using messages with happy and sad *emoticons*. An *emoticon* is a representation of a facial expression such as a smile or frown, formed by various combinations of keyboard characters and used in electronic communications to convey the writer's feelings or intended tone.

2.3 Short Text Clustering

The problem of streaming text clustering is particularly challenging in the context of text data because of the fact that the clusters need to be continuously maintained in real-time. Text clustering plays an important role in application areas such as text mining, information retrieval and search engines. However, traditional text clustering methods cannot be used directly for short text messages like *Twitter* messages because short texts often do not provide sufficient statistical information for effective similarity measures, and abbreviations are widely used in a message.

Rangrej et al. in [84] compare three different clustering techniques for short text documents including K-Means clustering, Singular Value Decomposition (SVD) and Affinity propagation. For K-Means, the authors use two variations of distance measures (i.e., derived from Cosine based similarity and Jaccard similarity). The number of clusters is manually selected with the minimum error. The SVD-based method and its relatives can be used in the topic identification of documents [19]. Affinity propagation is a graph-based algorithm. Each node represents a document and the edges represent the similarity among the messages. Based on the observations, the graph-based approach using affinity propagation performs best in their experiments.

Banerjee et al. in [6] propose a method to improve the accuracy of clustering short texts by enriching the representation with additional features from *Wikipedia*[2]. The titles of the retrieved *Wikipedia* articles are used as additional features. *Wordnet*[3] is an external resource which has been used for sense disambiguation of query terms, and then has had added synonyms of query words to expand the query [5, 23, 62]. Hotho et al. in [34] and Wang et al. in [111] propose to incorporate synonyms from *Wordnet* into text representation and they show that the extra features can improve text clustering quality. However, expanding every term can increase processing time and the number of terms. It may also add noise to the term vector and can decrease the performance of the clustering method.

Concept similarity based had been proposed for text classification [81] and text clustering [35, 38]. Hu et al. in [35] propose a framework to leverage the hierarchical, synonymy and associative semantic relations from *Wikipedia*'s thesaurus. *Wikipedia* concept sets are used to enhance clustering. This

[2]http://en.wikipedia.org/
[3]http://wordnet.princeton.edu

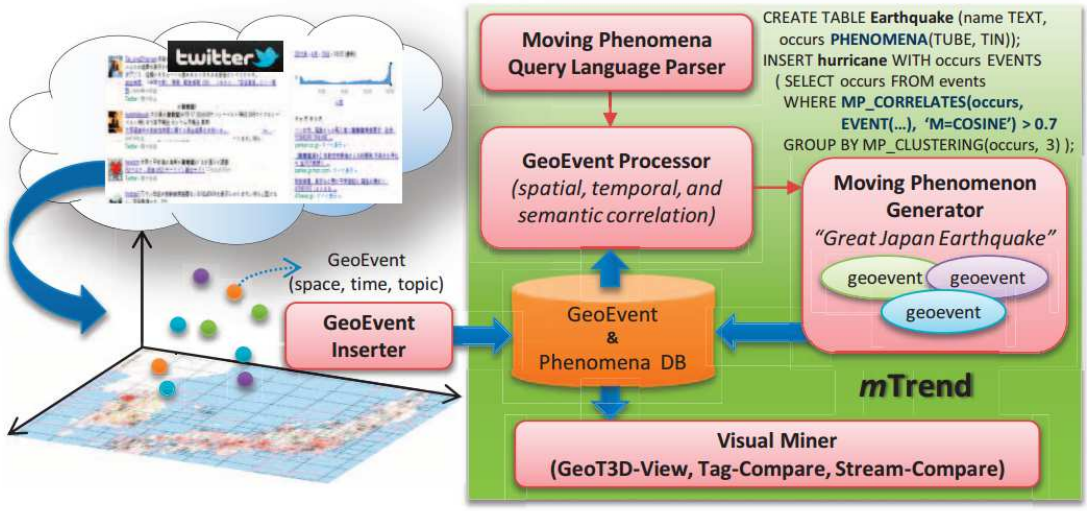


FIGURE 2.8: System architecture of *mTrend* [47].

approach need to build a concept thesaurus based on the data extracted from *Wikipedia* by themselves. Huang et al. in [38] present an approach to cluster documents using a *Wikipedia*-based concept representation. First, these authors create a concept-based document representation by mapping the terms and phrases within documents to their corresponding articles (or concept) in *Wikipedia*. Then, a similarity measure to evaluate the semantic relatedness between concept sets for two documents is proposed. However, *Wikipedia* itself is not a structured thesaurus such as *Wordnet* which makes it difficult to handle the problems of synonymy and polysemy. Moreover, both works focus on long documents rather than short messages like micro-blog messages.

2.4 Event Tracking in Social Networks

Sakaki et al. in [91] present real-time event detection by using *Twitter messages* associated with time and geographic location information to detect event occurrences such as earthquakes and typhoons. The *mTrend* approach is proposed by Kim et al. in [47] to analyse the continuous spatio-temporal trends and their movements in a space-time domain. Their work is similar to the work in [68] but these authors focus on analysing the continuous spatio-temporal trends and their movements in a space-time domain. They employ their moving-phenomena data model as proposed in [46] to represent topic movements from the *tweets*. The system architecture is presented in Figure 2.8. However,

none of these approaches can capture the evolution of events. Also, events may evolve in very different patterns such as transformation, dissolution, merging or splitting. So it is hard to define a generic model for all possible events. Furthermore a general life-cycle of events such as “emerging”, “developing”, and “concluding” may not be able to reflect the complexity of event tracking.

Lin et al. [60] present a framework for generating storylines from micro-blogs for user input queries. A dynamic pseudo relevance feedback language model is presented to retrieve relevant *tweets* given an event query. A graph-based optimization problem is applied to solve the problem of storyline generation. Lee et al. in [52] group posts into event networks and track six types of evolution patterns. First, the authors extract keywords from a social stream and transform them into an evolving post network by measuring pairwise post similarity. Then, they apply density-based clustering to identify events. Evolution patterns of events are tracked incrementally. Finally, event results are ranked and presented to users. However, the heuristic method for tracking the relationship of events may not effectively discover causality between news articles.

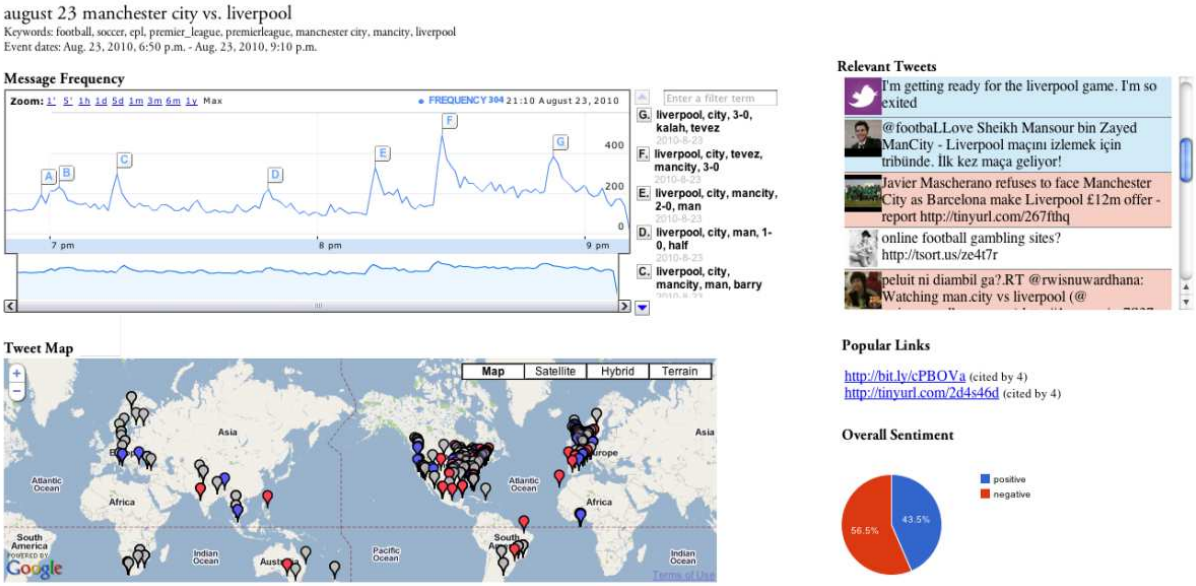


FIGURE 2.9: The *TwitInfo* user interface [66].

TweeQL is introduced by Marcus et al. in [65]. It is a query language that allows *tweets* to be queried by exposing fields such as location and text. Marcus et al. in [66] also introduce a system for visualizing and summarizing events on *Twitter* in real-time, namely *TwitInfo*; this can be seen in Figure 2.9. Events are extracted by identifying temporal peaks in message frequency from *Twitter*

messages containing keywords specified by the user. However, this approach does not answer whether these messages are related or not to the given event. To the best of our knowledge, every event is developed in a unique way and is transitionally coherent to a set of keywords at different stages of its life time. So the only restriction should be its time of emerging and its time of finishing. In other words, we should monitor an event not by any pre-specified patterns, but by the topics that are consequently related to the event and watch their changes within the given period of time.

Event identification and tracking in social media streaming data is proposed in [113]. Weiler et al. use a sliding window model to extract events and the context of events in real-time. Their approach is based on monitoring shifts in the inverse document frequency (IDF) of terms. An event term is identified if all shift values are higher than the corresponding average values. The top k co-occurrence terms of the event term are extracted to summarize the context around an identified term.

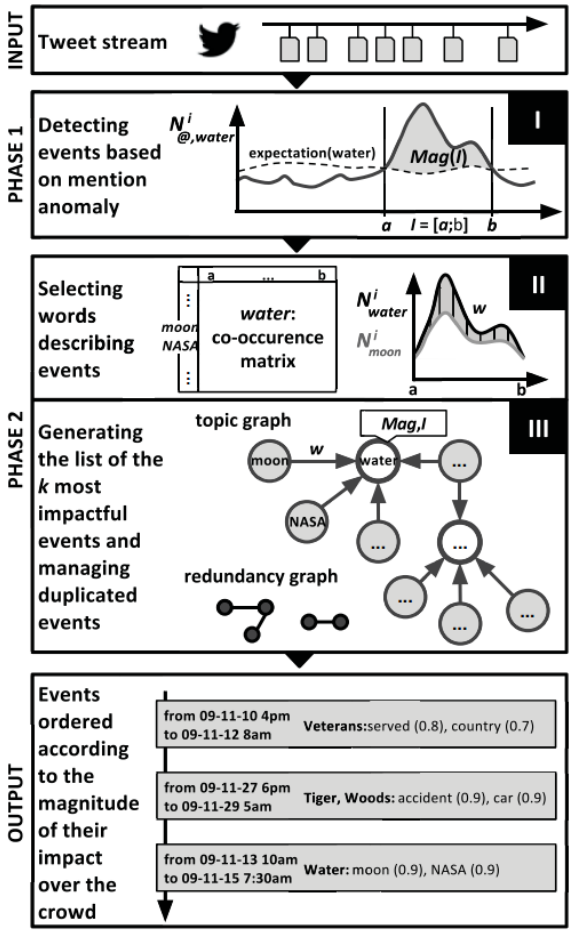


FIGURE 2.10: The overall flow of *MABED* method [31].

Guille and Favre in [31] propose Mention-Anomaly-Based Event Detection (MABED) and tracking in *Twitter*. Their method relies on three components: (1) event detection based on mention anomaly, (2) words selection to describe each event and (3) the top k events selection. The overall flow is shown in Figure 2.10. The authors compute the anomaly of a word for a given sequence of time-slices. The set of candidate keywords for describing an event is selected based on the p highest co-occurrence counts, where p is a parameter manually fixed by authors. The list of top k event is created by using graph structures; the topic graph and the redundancy graph.

Recent work in the analysis of world-wide event evolution is proposed by Huang et al. in [39]. They introduce a Finding Topic Clusters using Co-occurring Terms (FTCCT) algorithm to automatically generate topics from short messages. An Event Evolution Mining (EEM) algorithm to discover hot events and their evolutions is developed. Importantly, a discrete term in their approach belongs to only one topic in a corpus. However, their experiments are conducted on an English short text corpus (i.e., news titles from 157 countries). These news titles contain grammatical, syntactical, and stylistic standards where the writing used has a different style from that used in the micro-blog message.

2.5 Sentiment Analysis on Social Networks

There are several research papers discussing sentiment analysis via lexicon-based approaches [15, 20, 37, 43, 69, 73, 110]. Ding et al. in [20] introduce a holistic lexicon-based approach to solving this problem by exploiting external evidences and the linguistic conventions of natural language expressions. This system identifies semantic orientations of opinions as expressed by reviewers on product features from *Amazon.com*. However, the writing used has a different style from that used in the original micro-blog message. Joshi et al. in [43] propose the combination of an *Emoticon-based* and a *Lexicon-based* sentiment predictor for micro-blogs. They use four different sentiment-based lexicon resources (i.e., *SentiWordNet*, *Subjectivity lexicon*, *Inquirer* and *Taboada*) to identify users' opinions. Meng et al. in [69] present an entity-centric topic-based opinion summarization framework in *Twitter*. The topic is detected from *hashtags*-human annotated tags for providing additional context and metadata to messages. Target-dependent sentiment classification is used to identify the sentiment orientation of a message.

Recent researches in the field of political sentiment analysis are presented by Ringsquandl et al. in [88] and Wang et al. in [110]. A similar work to our approach is introduced in [88]. This work studies the application of the Pointwise Mutual Information measure to extract relevant topics from *Twitter* messages. Unsupervised sentiment classification is proposed. The semantic orientation of the word is the most probable class (positive, negative, neutral) of each opinion word according to *synsets* (i.e., synonym) in *WordNet*. The final aspect-level sentiment is determined by a simple aggregation function which sums the semantic orientation of all words in the message that mentions the specific aspect. However, the works mentioned above do not consider the case of sarcastic messages.

Sarcasm identification has been studied in only a few studies. Tsur et al. in [104] propose a semi-supervised approach for sarcasm identification in product reviews. These researchers use *Pattern-based* and *Punctuation-based* features in order to train the classifier. In [28], González-Ibáñez et al. present an empirical study on the use of lexical and pragmatic factors to distinguish sarcasm from positive and negative sentiments expressed in *Twitter* messages. These researchers compared the performance of automatic and human classification in different studies, however, no one has incorporated a sarcasm recognition module in sub-event detection and sentiment analysis.

Chapter 3

Location-based Emerging Event Detection in Social Networks

3.1 Problems and Challenges

With the increasing number of real-world events that are originated and discussed over social networks, event detection is becoming a compelling research issue. However, the traditional approaches to event detection on large text streams are not designed to deal with a large number of short and noisy messages. Our interest is to understand **where**, **when** and **what** an event is happening (emerging) so to detect its occurrence via the real-time monitoring on the social networks. Events are often location-sensitive and knowing where an event occurs is as important as knowing when it happens. More specifically, our research is focused on the emerging **"hotspot"** events, that is, emerging events with respect to the locations and the participants of the events. We define that a hotspot event is a tuple (*location, time, topic*) that a social network user is associated with through the posting of a micro-blog.

In this chapter, we proposed an approach called Location Sensitive Emerging Event Detection (*LSED*) in social networks. This approach is used to detect emerging hotspot events from micro-blog messages that can help government or organizations prepare for and respond to unexpected events. With the large range of events discussed on social networks, we may want to know how many emerging hotspot events are likely to happen. However, it is difficult to effectively and efficiently

process a large number of noisy messages. The research challenges of our study are: (1) how to effectively detect events in terms of keywords in micro-blogs? (2) how to detect hotspot events (i.e., associate the message-mentioned location(s) to an event)? and (3) how do we know a hotspot event is emerging?

The problem that we address in this work is how to identify emerging events with location sensitivity from a given set of micro-blog messages. We consider a set of messages where each message is associated with an event. However, due to the characteristics of micro-blog messages, several issues are listed as follows:

- People share various types of content such as conversation topics, advertisements, events, opinion, and others. Our goal is to detect only emerging hotspot events that are happening in particular area.

- The weighting scheme of micro-blog messages should differ from traditional methods because the micro-blog message is very short and often does not provide sufficient information. Abbreviations are also widely used in a message.

- With the large range of events discussed on social networks, we do not know the number of events in advance. Traditional clustering methods like K-Means technique should determine the fixed number of clusters. It is unsuitable for the real world system when dealing with dynamic topics or events.

We identify our emerging hotspot event detection problem as presented in Figure 3.1. Consider a time-ordered micro-blog messages M, each message includes *message id, message content, creating time, user id, user location* and/or *tweet location* (i.e., *geo-tagged*). Given a sliding windows size s and the number of previous time slots p at time t, our task is to detect emerging hotspot events via the real-time monitoring of micro-blog messages M.

A sliding window manager is used to keep track of messages arriving in the approach. The size of the sliding window depends on the user-given preference such as *1 hour, 2 hours* or *6 hours*. The number of previous time slots needs to be specified. It is used to define history data for computing the emergence of the event. Our approach consists of three core methods:

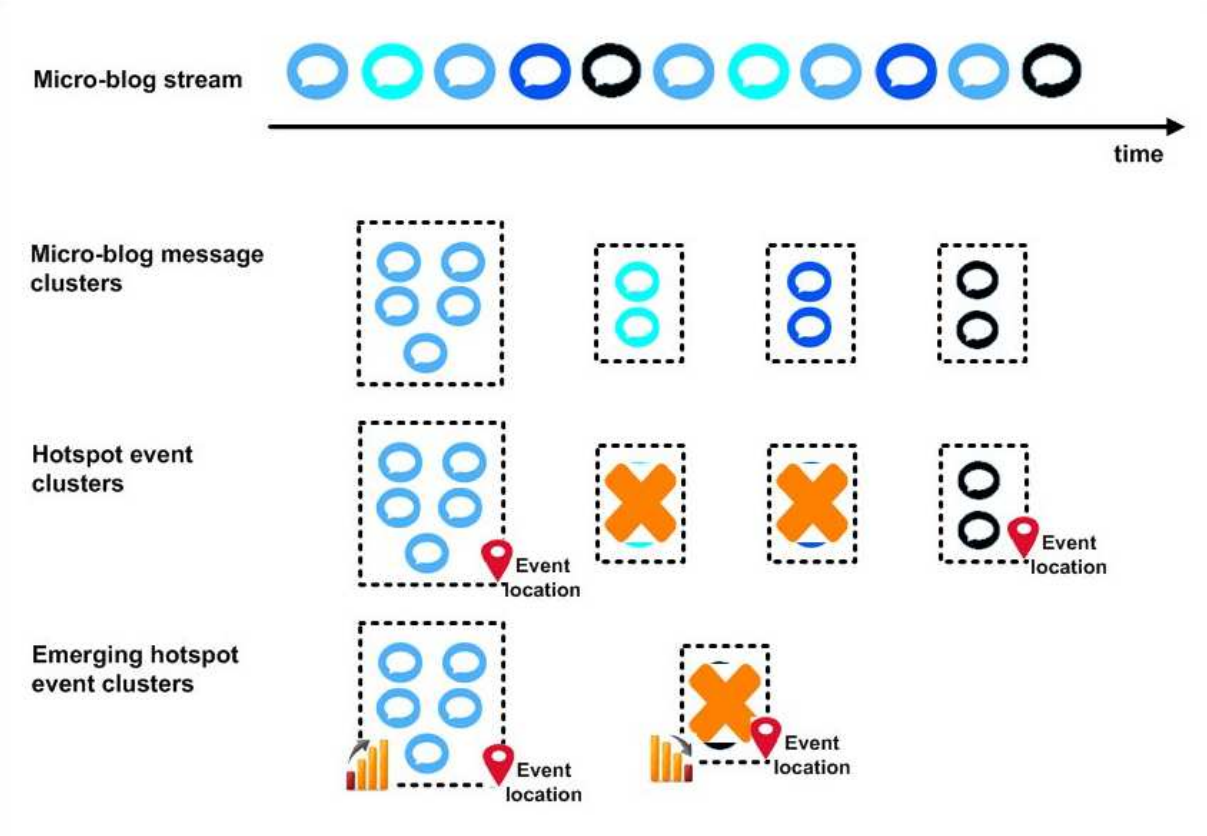


FIGURE 3.1: Conceptual diagram of emerging event detection with location sensitivity.

- **Micro-blog message clustering:** given a time-ordered micro-blog messages M, we aim to automatically group messages M into the same cluster such that each cluster is associated with only one event.

- **Hotspot event detection:** given the clusters C from previous stage, the hotspot event is identified by finding the strong correlation between user locations and event locations.

- **Emerging hotspot event detection:** given the hotspot event clusters C_h, our task is to detect an emerging hotspot event by observing changes in the popularity of event (i.e., the number of messages in the event cluster).

Next, we introduce the notations and definitions for event detection in social network streams. We assume that the structure of the social network is denoted by the graph $G = (V, E)$. The node set is denoted by V to represent users in social networks and edge set is denoted by E to represent the relationship between users. A document stream, *Twitter* message, is a time-ordered sequence of messages. Within the document stream, social network messages are always processed chronologically,

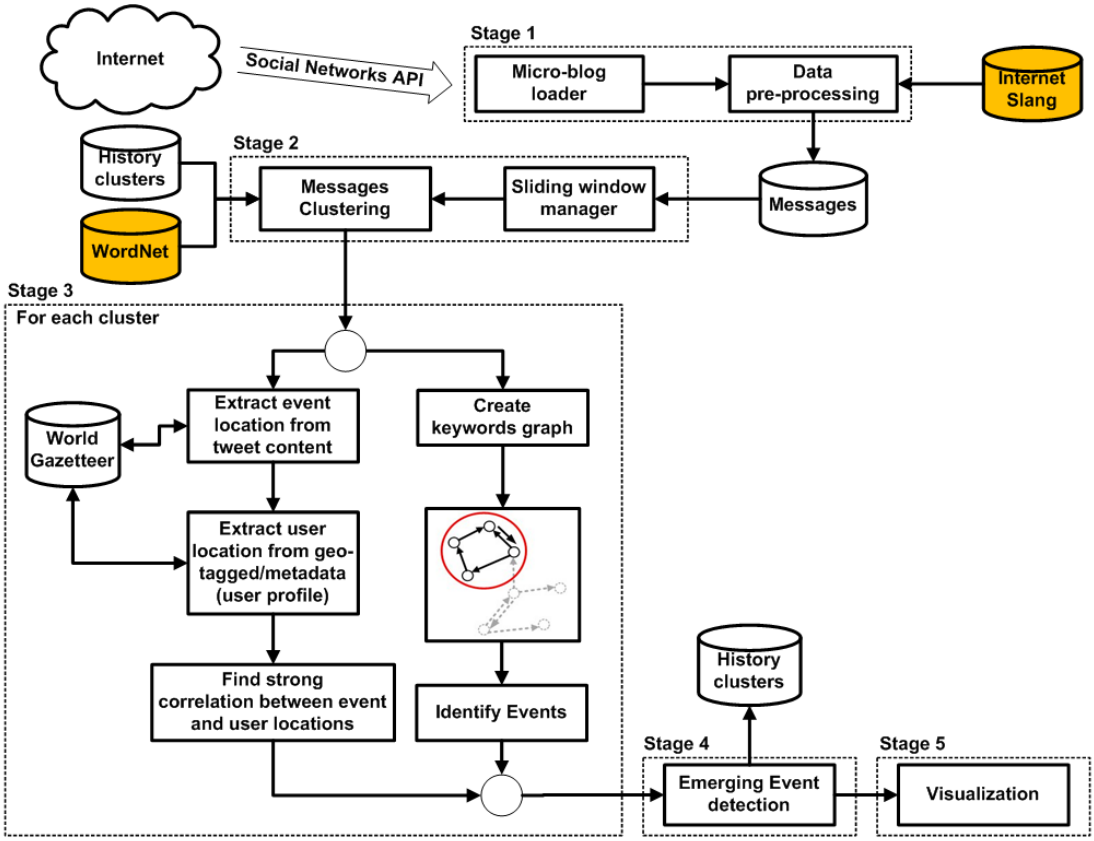


FIGURE 3.2: Architecture of LSED system.

as they are posted. In our work, we define the definition of document stream as follows:

Definition 3.1. *A "document stream" is a continuous and time-ordered sequence of messages* $M = \{m_1, m_2, ..., m_n\}$ *such that each message* m_i *contains text content* c_i*, user profile (i.e., the sender of the message), and/or re-tweeter. The message* m_i *contains the origination node* $v_i \in V$ *which is the author of the message. The message* m_i *contains a set of one or more re-tweet nodes* $R_i \subseteq V$*, which correspond to all re-tweeter of the message from node* v_i*. Therefore, the message* m_i *is represented by the tuple*(c_i, v_i, Ri)*.*

3.2 Emerging Event Detection with Location Sensitivity Approach

In order to provide a complete coverage of the system, we produced an approach which has five stages as presented in Figure 3.2. Firstly, the pre-processing is performed to remove stop words and irrelevant data. Secondly, we propose a clustering approach to automatically group the messages into events. Thirdly, we propose a hotspot event detection method. Fourthly, emerging hotspot event

detection is performed. Finally, we develop a visualization model for representing emerging events. The following information provides details of each stage.

3.2.1 Data Pre-processing for Events Detection

Pre-processing

In dealing with micro-blog messages, the message is short and often noisy. In order to improve the quality of our dataset and the performance of the subsequent steps, the pre-processing was designed to ignore common words that carry less important meaning than keywords and to remove irrelevant data e.g., *re-tweet* keyword, web address and message-mentioned username. A micro-blog loader is developed to collect the *Twitter* data from public users via the *Twitter* API service. Web addresses and single character words are removed from the messages. We also remove the keyword RT(*"ReTweet"*) and the message-mentioned username, such as *"@Sayan"* from the message. Next, the messages are converted into lower case.

Slang conversion

In general, *twitter* messages are informally written and often contain grammatically incorrect text with misspellings and abbreviations. For example, the word *"tomorrow"* can be *"2ma", "2maro", "2mmrw", "2mo", "2mora", "2moro", "2morow", "2morro", "2morrow", "2moz", "2mozz", "2mro", "2mrw", "2mw",* or *"2mz"*. In a traditional bag-of-words model each slang word is treated as a different feature but in fact, they should represent the same word. Our aim is to convert the abbreviations into proper English words to improve the performance of the message similarity in the next step.

We downloaded the Internet slang dictionary from http://www.noslang.com and stored this in the database. Examples of slang words are shown in Table 3.1. For each term except *url, hashtag* and *mentions* in *Twitter* message, we search for it in the slang dictionary database and convert it into a proper English word. In order to handle extensions like *"goooood", "gooooodddd"* and *"yesssss"*, we replace consecutive occurrences of the same letter (if more than three occurrences) with a single or double identical letters. For example, the word *"goooood"* will be replaced with *"god"* and *"good"*.

TABLE 3.1: Examples of the Internet slang dictionary.

proper English word	slang words
between	b/t, b/w, btwn
busy	bizi, buszay, bz, bzy, bzzy
hacker	h4x0r, h4xor, h4xr, h4xx0rz, h4xxor, hax0r, haxer, haxor, haxxor, haxxzor, haxz0r, haxzor, xor
hate	h8, h8t, h8te, hait, heyt
tomorrow	2ma, 2maro, 2mmrw, 2mo, 2mora, 2moro, 2morow, 2morro, 2morrow, 2moz, 2mozz, 2mro, 2mrw, 2mw, 2mz
tonight	2night, 2nite, 2nyt

Finally, the stop words are removed and all words are converted into a seed word (stemming word) by using Lucene 3.1.0 Java API[1]. All messages after being pre-processed are stored in the database. In order to illustrate the performance improvement, the experiment results will be shown in Section 3.3.

3.2.2 Text Stream Clustering for Event Detection

Micro-blog Message Clustering

The problem that we address in this section is how to group a set of micro-blog messages into a cluster. We consider a set of messages where each message is associated with an event. With the large range of events discussed on social networks, we do not know the number of events/clusters in advance. Traditional clustering methods like K-means technique should determine the fixed parameter of k (i.e., the number of clusters). It is unsuitable for the real world system when dealing with dynamic topics or events. In other words, our approach requires no prior knowledge of the number of events. Therefore, hierarchical clustering is used in our approach.

In this stage, we aim to automatically group messages into the same event. We also need a fast and efficient message clustering system to overcome the problem of the high arrival rate of messages. In order to deal with the high incoming rate of messages, we use a sliding window manager to keep track of messages arriving in the system. The size of the sliding window can be defined as the number of messages or time interval. In our case, we use time intervals such as one hour, two hours or one day depending on the user-given preference. Additionally, the number of previous time slots needs to

[1] http://lucene.apache.org

be specific because it is not necessary to consider the complete usage history of data to compute the emergence of the event. We formally define the text message clustering problem as follows:

Definition 3.2. *A document stream* $M = \{m_1, m_2, ..., m_n\}$ *is continuously grouped into clusters* $\{C_1, C_2, ...\}$ *such that each message* m_i *belongs to at most one of the cluster* C_j*. The messages are assigned to the different clusters by using a similarity function. The new message is compared with the previous clusters for a particular time period.*

In order to find the best representation of term weight for *tweet* messages, we compare four different term weight formulas (i.e., Term frequency, Augmented Normalized Term Frequency[92], TFIDF[92], and Smooth-TFIDF[76]). We also compare four different similarity functions (i.e., Jaccard index, Euclidean distance, Manhattan distance and Cosine similarity) for finding the best similarity function. To evaluate an effective clustering method, we manually label 15,852 *tweets* into 17 topics. We evaluate our algorithm by using pair-wise precision, recall and F1-score. Our preliminary experiments will be shown in Section 3.3. The clustering method performs well when using the augmented normalized term frequency and cosine similarity function. Therefore, we calculate the weight $w_{i,t}$ of term t in message i by using augmented normalized term frequency:

$$w_{i,t} = 0.5 + 0.5 \times \frac{tf_{i,t}}{tf_i^{max}} \tag{3.1}$$

where $tf_{i,t}$ is the term frequency value of the term t in message i, tf_i^{max} is the highest term frequency value of the message i.

The cosine similarity function is used to calculate the similarity between the existing cluster and the new message:

$$ContentSim(m, c) = \frac{\sum_i (w_{m,t_i} \times w_{c,t_i})}{\sqrt{\sum_j w_{m,t_j}^2} \times \sqrt{\sum_j w_{c,t_j}^2}} \tag{3.2}$$

where m is a message, c is a cluster centroid, and w_{m,t_i} is the weight of term t_i in message m.

We use a clustering method called leader-follower clustering [21]. Message clustering is executed when the timestamp of the coming message is greater than the sliding window size. Our approach keeps history clusters within previous time slots to decrease time of computation. Every new message is compared with previous clusters. The algorithm creates a new cluster for the message if there is

Algorithm 5: $TextClustering(M, C_h)$

Input: M : list of messages in current sliding window, C_h: history clusters
Output: C : all clusters

```
1  topWords = getTopKeywords(M, 20%);
2  while m ∈ M do
3      getm;
4      m = expandKeywordByWordNet(m, topWords);
5      c_m = getConceptByWordNet(m, topWords);
6      MSim = find most similar cluster by using Eq. 3.2;
7      if MSim ≥ threshold then
8          MSim.addMessage(m);
9          //assign message m to the most similar cluster
10     else
11         newCNew;
12         CNew.addMessage(m);
13         // create new cluster (CNew) and assign message m to CNew C_h = C_h ∪ CNew;
14 C = C_h;
15 merge similar clusters by using Eq. 3.4;
16 return C;
```

no cluster whose similarity to the message is greater than threshold. We use a centroid representation of the cluster because centroid is agglomerative, using the mean, which trades memory use for speed of clustering. In the final step, we find the most similar pair of clusters and merge them into the same cluster. In order to find the most similar cluster, we calculate cosine similarity and conceptual similarity between two clusters. If the combined similarity value exceeds the margin threshold, we merge them. The clusters which contain more than one message are selected as candidate event clusters. The short text clustering algorithm is shown in Algorithm 5.

Keywords expansion via WordNet

In micro-blog messages, it is possible that people may use different words when they are talking about the same thing, for example, the word *"earthquake"* can be *"quake", "temblor"* or *"seism"*. The example messages are shown below.

Message 1: *"OMG!! **earthquake** attacks indonesia!! #indonesia"*

Message 2: *"**#quake** is still happening in #Indonesia :(I wonder when it ends."*

It can be improved by better utilizing the semantic information available from lexical resources such as *WordNet*[72]. In order to improve the performance of short text clustering, we use the synonym expansion method for improving the accuracy of micro-blog message clustering focused on enriching short text representation. We use the keywords list with *WordNet* to expand the initial text representation. *WordNet* assigns words of English language to sets of synonyms called "*synsets*". It separates the data into four databases associated with the categories of verbs, nouns, adjectives and adverbs. In this research, we only expand the nouns and verbs. We downloaded the *WordNet* Dictionary database[2] and used the *Java* library called the *MIT Java WordNet Interface*[3] (JWI) to access the *WordNet* electronic dictionary. To complete the above example, the keyword *"earthquake"* will be expanded as follow.

V1: {*earthquake [quake, temblor, seism] (0.75), attack (0.75), indonesia (1.00)*}

V2: {*quake [earthquake, temblor, seism] (1.00), happen (1.00), indonesia (1.00),*

wonder (1.00), end (1.00)}

The vectors $V1$ and $V2$ have a cosine similarity of 0.837, which is higher than the value of 0.307 achieved without synonym expansion. However, not all of the keywords are expanded. Expanding every term can increase processing time and the number of terms. It may also add noise to the term vector. According to our experiments, the keywords list consists of the top x percent keywords based on their augmented normalized term frequency score. Our tests show that when $x = 20$ it renders the best performance. Therefore, in this work, we select 20 percent of the top keywords to be expanded.

Concept similarity computing for cluster merging method

Bag of words (*BoW*) and term frequency-inverse document frequency (*TF-IDF*) are commonly used for document classification, document clustering. However, words are limited for short texts like micro-blog messages. The clustering performance is relied on the similarity measure of message pairs. Therefore, finding an precise similarity measure is important for improving short messages clustering performance.

In order to merge similar clusters together to reduce the number of duplicate clusters (or events),

[2]http://wordnet.princeton.edu

[3]http://projects.csail.mit.edu/jwi/api/

we propose cluster similarity measure with a conceptual similarity. Enriching the term with concepts by introduce more general concepts extracted from *WordNet* (called *Hypernym*) can help identifying related topics. A cluster c is represented by a combination of concept sets of all messages in cluster c. However, not all the terms are extracted their concepts. Extending every terms may add noise to our approach. Therefore, the 20 percent of the top terms (selected from Section 3.2.2) are extracted the concepts of cluster c. Given two clusters c_i and c_j, the Jaccard similarity function is used to calculate the conceptual similarity between two clusters. Their conceptual similarity is defined as:

$$ConceptSim(c_i, c_j) = \frac{|C_{c_i} \cup C_{c_j}|}{|C_{c_i} \cap C_{c_j}|} \tag{3.3}$$

where C_{c_i} is a concept set of a cluster c_i and C_{c_j} is a concept set of a cluster c_j. We then define the overall similarity between two clusters as a linear combination of the message content similarity and conceptual similarity.

$$CSim(c_i, c_j) = (1 - \lambda)ContentSim(c_i, c_j) + \lambda ConceptSim(c_i, c_j) \tag{3.4}$$

where λ is a parameter that we set to 0.7 based on our experiments. However, if the concept of a given cluster cannot identify we only compute content similarity (i.e., $CSim(c_i, c_j) = ContentSim(c_i, c_j)$). For the function $ContentSim()$, we use a centroid of the cluster to compute the content similarity between two clusters. The clustering component may affect the final detection results so text clustering evaluation will be assessed. The experiment results of this stage will be shown in Section 3.3.

3.2.3 Hotspot Event Identification

From the previous stage results, all clusters cannot be assigned as event clusters because they can be private conversations, advertisements or others. In this step, we focus on how to identify the hotspot event clusters. According to event definition, a cluster will be assigned as the event cluster if we can extract the event location from the message contents in the cluster. A cluster also will be identified as the hotspot event if there is strong correlation between the event location and the user location. In order to understand what is going on we also need to extract a set of keywords to represent event

topic. We formally define a hotspot event identification problem as follows:

Definition 3.3. *A hotspot event can be regarded as* $E(Topic, userLocation, eventLocation)$*, where* $userLocation$ *is a location where the message is sent from and* $eventLocation$ *is a location where the event occurs. A cluster* C_i *will be assigned as the event when there is a strong association between event location and user location.* $eventLocation$ *has a high frequency (appears in most messages in cluster* C_i*).* $Topic$ *is represented by the set of co-occurring keywords in cluster* C_i*.*

The tasks are divided into two parts; to find a correlation between the event location and user location in the cluster, and to extract the event topic.

Find a correlation between the event location and user location

In order to calculate a correlation score, for each cluster we need to extract the user locations and event location first. Then, we calculate the correlation scores between user locations and event locations for identifying the hotspot event.

User location extraction: we can extract user location from the *geo-tagged* information and the user profile. The *geo-tagged* information is generated from smart phone applications while the other one is the free format text which the user fills in the user profile. For those users who can post messages from different locations, for a given message we use *geo-tagged* information to locate user location firstly, because it can provide the precise location of the user. In order to convert a latitude/longitude pair into an address, we use Google Maps API[4]. If *geo-tagged* information is not available we use user location in the user profile to query the *Gazetteer* database, the database of geographic locations, for acquiring the locations address. Finally, if neither of them is available we set user location equal to "World". For the *Gazetteer* database, we downloaded a list of geographic locations from *GeoNames*[5] and stored a local copy of the *Gazetteer* in a database. The granularity level is defined as *"Country> State>City>PlaceName"*. Examples of user and *tweet* locations conversion are presented in Table 3.2.

Event location extraction: we find all terms or phrases which reference geographic location (e.g. country, state and city) from *tweet* contents. Since the location extraction from text is one of the

[4]https://developers.google.com/maps/documentation/geocoding/
[5]http://www.geonames.org

TABLE 3.2: Examples of user and *tweet* locations conversion from *Twitter*.

Message id	User ID	User location (from user profile)	Geo-tagged (tweet location)	Convert result
1	26269xxxx	Brisbane, Australia		AU>QLD>Brisbane
2	26269xxxx	Brisbane, Australia	-33.8705, 151.21	AU>NSW>Sydney> David Jones
3	21592xxxx	iPhone: -27.469482, 152.987442		AU>QLD>Brisbane> Toowong
4	7457xxxx	Brisbane		1) AU>QLD>Brisbane or 2) US>CA>Brisbane
5	14372xxxx	Wherever!!		"World"
6	1611xxxx	London.... Sydney, Australia		"World"
7	35124xxxx		-27.3885, 153.1199	AU>QLD>Brisbane> Brisbane Airport

challenging problems of this research area, in this paper we simply extract the message-mentioned locations via *Named Entity Recognition (NER)*. We use the *Stanford Name Entity Recognizer* [45] to identify locations within the messages. We also use the *Part-of-Speech Tagging* for *Twitter* which is introduced in [27] to extract proper nouns. We use an extracted terms query into the *Gazetteer* database to obtain candidate locations of the event. We find the most probable location of the event using the frequency of each location in the cluster. Based on our observation in Chapter 1, the location which has a highest frequency is assigned as the event location if the frequency exceeds the threshold otherwise "World" will be assigned. The threshold is defined as the percentage of the location is mentioned in the cluster to avoid small number of location. The message mentioned location examples of a given event are shown below:

M1: *huge earthquake in **chile**, cant believe this is happening again :[*

M2: *God Bless all those who were hurt in the earthquake and aftershocks.*

M3: *OMG! Earthquake in **Chile**... **Hawaii** under tsunami warning...*

M4: *Did anyone hear about the enormous 8.8 earthquake in **Chile**? Holy crap!*

There are two locations mentioned in the messages, however ***Chile*** is the most mentioned location. Thus, ***Chile*** is identified as event location in this event.

Finding the correlation between event location and user location: a correlation score is computed by comparing the level of location granularity. We assign scores for each level if both of

them have the same value. The equation is shown below:

$$CorrelateScore = \alpha 1(F(uCountry, eCountry)) + \alpha 2(F(uState, eState)) \\ +\alpha 3(F(uCity, eCity)) + \alpha 4(F(uPlace, ePlace)) \quad (3.5)$$

where $\alpha 1$ - $\alpha 4$ are the weight of granularity levels, $\alpha 1 = \alpha 2 = \alpha 3 = \alpha 4 = 0.25$, $uCountry, uState$, $uCity$ and $uPlace$ are user location, $eCountry, eState, eCity$ and $ePlace$ are event location, and $F(x, y) = 1$; if $x = y$ and the higher granularity level has the same value otherwise $F(x, y) = 0$.

To identify which cluster is a hotspot event, the $LocScore$ is computed. The range of $LocScore$ is 0 to 1. It will be used to compute emerging score in the next section and the top k ranking emerging hotspot events will be selected. The $LocScore$ of cluster c is defined as:

$$LocScore_c = \frac{\sum_{u \in U} CorrelateScore_u}{|U|} \quad (3.6)$$

where $|U|$ is the number of users who post messages in cluster c.

Event Topic Extraction

In order to understand what the event cluster is about, we need to find the set of keywords to represent the event topic. Our intuition is that keywords co-occur when there is a meaningful topical relationship between them. To extract the set of co-occurring keywords, firstly we create a directed, edge-weighted graph. The edge is created if the correlation weight between the two terms exceeds the threshold. The threshold is defined as the average of correlation weights in the cluster.

We adopt the smoothed correlation weight function which is introduced in [90], to calculate the semantic correlation weight between terms. The function is shown below:

$$c_{k,z} = log(\frac{(n_{k,z} + \frac{n_k}{N})/(n_z - n_{k,z} + 1)}{(n_k - n_{k,z} + \frac{n_k}{N})/(N - n_k - n_z + n_{k,z} + 1)}) \quad (3.7)$$

where n_k is number of posts containing term k, n_z is number of posts containing term z, $n_{k,z}$ is number of posts containing the terms k and term z, while N is the total number of posts.

We identify the event topic by extracting the *Strongly Connected Components (SCCs)* from the

graph. In the case of *SCCs*' extraction, if the number of *SCCs* is more than one sub-graph we calculate the sum of edge weights on each *SCC* sub-graph. The *SCC* which has the highest score is defined as an event topic. The example of even topic extraction is shown in Figure 3.3.

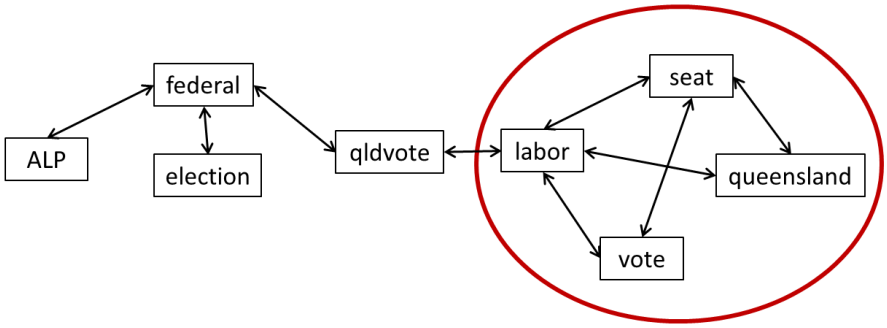


FIGURE 3.3: Example of event topic extraction.

3.2.4 Emerging Hotspot Event Detection

As our research interest is to detect emerging hotspot events, all events from the previous stage are not assigned as emerging hotspot events. According to our definition of emerging event, we find the event that has significantly increased in the amount of messages but has rarely been posted in the previous time slots. A burst in data is one technique for detecting emerging events. It is calculated by comparing the number of messages in the current time slot with the mean and the standard deviation of the number of messages in the previous time slots. Any data point which is higher than the sum of the mean and two standard deviations can be considered as an emerging point. Emerging event detection algorithm is shown in Algorithm 6.

In order to compute an emerging score for a given event's cluster we calculate the mean and the standard deviation of the number of messages in the previous time slots. If the number of messages in the current time slot is greater than the sum of the mean and two standard deviations, we calculate the emerging score of this event. The emerging score of the event e in the current slot is computed by the following equation:

$$EmergScore_e = (1 + LocScore_e) \times \frac{N_e}{(Mean_{prev} + 2SD_{prev})} \tag{3.8}$$

Algorithm 6: $EmergingEventDetection(C, list\_ulocs)$

Input: C: all hotspot event clusters, list_ulocs: list of user locations from all clusters
Output: list : list of emerging hotspot events

```
1 list = null;
2 forall the l ∈ list_ulocs do
3     forall the c ∈ C do
4         Mean_prev = calculateMeanPrevSlots();
5         // calculate mean of message frequency in previous time slots
          SD_prev = calculateStandardDevPrevSlots();
6         // calculate standard deviation of message frequency in previous time slots
          emergingScore = calculateEmergingScore(c);
7         if emergingScore ≥ 1.0 then
8             list.addCluster(c);
9 return list;
```

where $LocScore_e$ is the location score of event e, N_e is the number of messages of event e in the current time slot and $Mean_{prev}$ and SD_{prev} are the mean and standard deviation of the number of messages in the previous time slots of the given event, respectively. The $(1 + LocScore)$ is used to boot up $EmergScore$ in case the values of the second part in the Eq. 3.8 of two events are the same. The events that have stronger correlation between event location and user location will have a higher $EmergScore$ than other one. $EmergScore$ must be greater than or equal one because the number of messages in the current time slot must be greater than the sum of the mean and two standard deviations according to Eq. 3.8.

To detect emerging hotspot events in different location granularity such as state and city level, we firstly segment event clusters into user's location groups according to location granularity and follow all of the steps above.

3.2.5 Visualization

For usability and understanding issues of visualizing the model, we designed a dashboard to display an event. We use a motion chart[6] to represent a specific emerging hotspot event in different areas and the period of time. Examples of a motion chart is shown in Figures 3.4 and 3.5. Both figures present an emerging event called *"Tropical storm Debby"* in different States in the US during 21 June 2012

[6]http://code.google.com/apis/chart/interactive/docs/gallery/motionchart.html

to 27 June 2012. Figure 3.4 shows a given emerging hotspot event within a time period (*x-axis* is the number of messages were posted; *y-axis* is the number of users who post messages; colour represents location and bubble size is the emerging score). The bar chart is shown in Figure 3.5 where *y-axis* is the number of users who post messages or the number of messages was posted in each day.

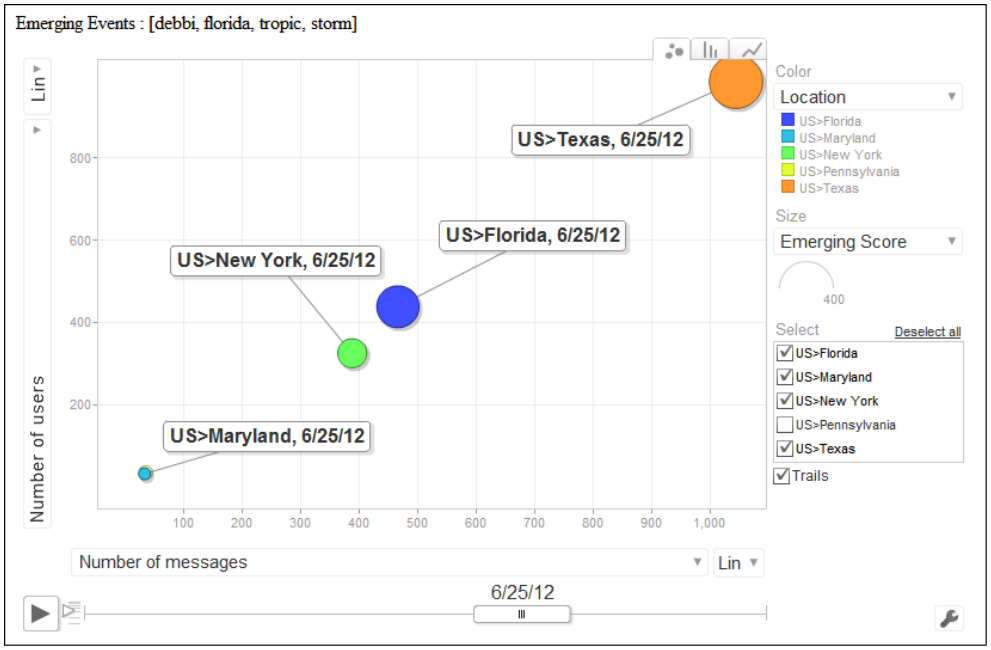


FIGURE 3.4: Example of a motion chart of Debby Storm event in different locations.

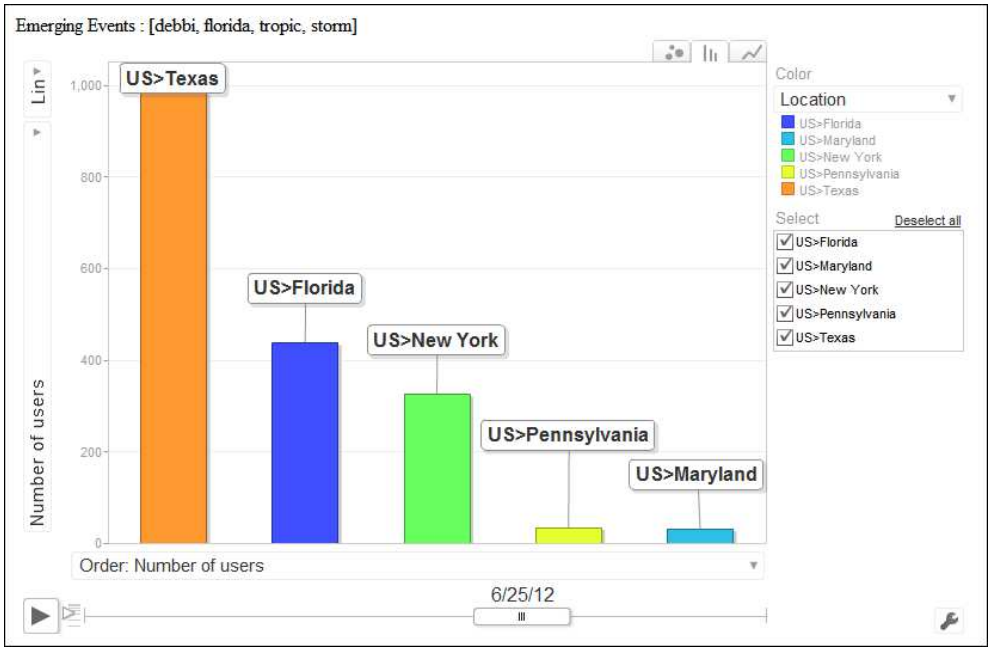


FIGURE 3.5: Example of a bar chart of Debby Storm event in different locations.

Google Map[7] is used to represent the top k emerging hotspot events for a specific location and time. The top five emerging events in the US on 25 June 2012 are shown in Figure 3.6. As we can see from the figure, the top two events are related to the same event in Florida State which were talking about *"Tropical storm Debby"*.

All emerging events in the US are presented via *Annotated Time Line Chart*[8], an interactive time series line chart with optional annotations provided by *Google API*. Events are shown in Figure 3.7 for each day (represented by letters A to Z). The number of messages is presented for each event. An event name is represented by a keywords list described in 3.2.3.

FIGURE 3.6: Geo-map of top five emerging events in the US on 25/6/2012.

3.3 Experiments and Results

To evaluate our approach, we firstly assess the clustering method because the clustering component may affect the final detection results of our approach. Next, we evaluate the performance of event detection results.

[7]https://developers.google.com/maps/documentation/javascript/examples
[8]https://google-developers.appspot.com/chart/interactive/docs/gallery/annotatedtimeline

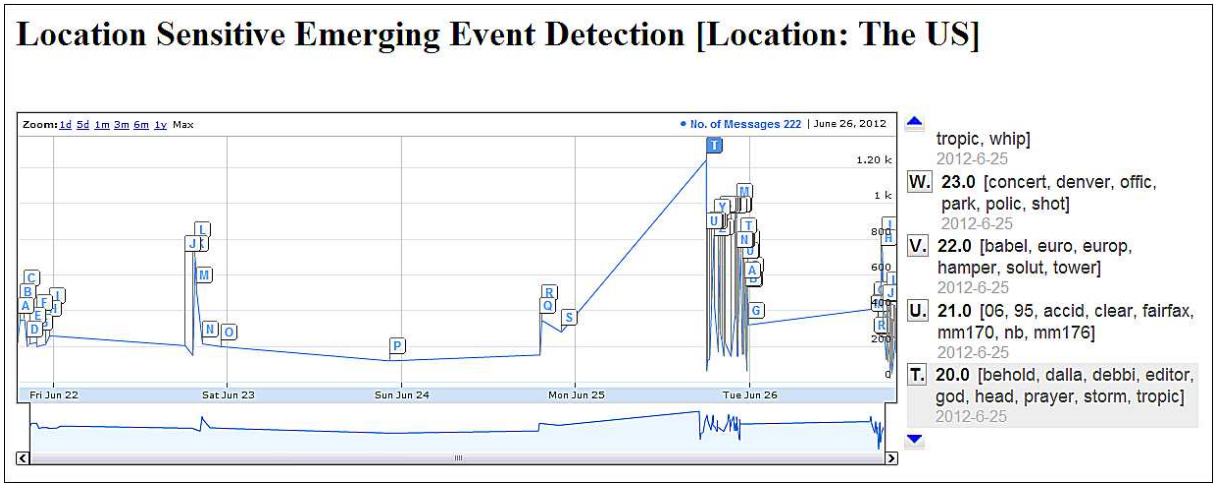


FIGURE 3.7: Annotated time line chart of emerging events in the US between 21-27 June 2012.

3.3.1 Clustering Method Evaluation

In order to find the best solution of micro-blog message clustering, we manually label 15,852 messages from *Twitter* which belong to 17 events. We evaluate our algorithm by using *Pair-wise Precision, Recall* and *F1-score*.

$$Pairwise_{precision} = \frac{|T \cap C|}{|C|} \tag{3.9}$$

$$Pairwise_{recall} = \frac{|T \cap C|}{|T|} \tag{3.10}$$

$$Pairwise_{F1Score} = \frac{2 \times Pairwise_{precision} \times Pairwise_{recall}}{Pairwise_{precision} + Pairwise_{recall}} \tag{3.11}$$

where T is the true clusters, C is system generated clusters, $|T|$ is number of pairs of messages that are in the same group in T, $|C|$ is number of pairs of messages that are in the same group in C, and $|T \cap C|$ is number of pairs of messages that are in the same group in both T and C.

Table 3.3 shows the clustering results with different term weights. As we can see from Table 3.3, using augmented normalized term frequency and slang conversion can effectively group micro-blog messages into the same cluster better than other methods.

We also try to improve our clustering method by using the synonym expansion and conceptual

TABLE 3.3: Clustering results compared against different Term weights with cosine similarity.

Method	Precision %	Recall %	F1 Score %
tweet contents with TF	98.90	19.30	32.40
tweet contents with Smoothed TFIDF	99.70	17.20	29.40
tweet contents with TFIDF	99.31	25.22	40.23
tweet contents with Augmented Normalized TF	55.09	45.82	50.03
tweet contents with Augmented Normalized TF + slang converting	71.49	97.13	**82.36**

TABLE 3.4: Clustering results with the different number of expanded keywords.

Percent of keywords expansion	Precision %	Recall %	F1 Score %
without expansion	71.49	97.13	82.36
10	97.69	89.01	93.15
20	97.64	91.90	**94.68**
30	97.66	91.72	94.60
40	97.23	87.07	91.87
50	97.22	85.25	90.84
60	95.44	88.49	91.83
70	97.11	83.60	89.85
80	97.32	83.23	89.73
90	96.96	77.63	86.23
100	96.65	74.56	84.18

similarity method from *WordNet*. We try to find out how many keywords we need to expand to obtain the best performance. Table 3.4 shows the clustering results with different numbers of keywords to be expanded. Our experiments show that when the top 20 percent of keywords are expanded it renders the best result. For concept similarity computing, Table 3.5 shows that the combination between content similarity and conceptual similarity can improve the clustering performance with *F1 Score* equal to 96.56%.

3.3.2 Event Detection Evaluation

In order to evaluate our approach, we use the search API from *Twitter* to collect the messages sent by users around the USA, from the dates 21 June 2012 and 27 June 2012 - this comprised 196,834 messages. Since no ground-truth labels are available for us on realistic events within the data collection period, we manually search local news from Google to check the events detected by our system.

TABLE 3.5: Cluster merging method performance.

Method	Precision %	Recall %	F1 Score %
without cluster merging	97.64	91.90	94.68
with cluster merging	97.53	95.61	**96.56**

It is impractical to manually label the overly large number of *tweets* in the dataset. We follow the definition of *Precision* used in [114] with minor changes, which is defined as follows:

$$Precision = \frac{detected\_realworld\_local\_events}{total\_detected\_realworld\_events} \tag{3.12}$$

However, *Recall* was not defined in [114] because it is not feasible to enumerate all the real-life events which happened in the dataset. Therefore, we follow the definition of *Recall* used in [56], which is defined as the number of distinct realistic local events detected from the dataset. Note that, more than one detected event can relate to the same real-world event, then they are considered correct in terms of precision but only one event is considered in counting recall.

$$Recall = no\_distinct\_realworld\_local\_events \tag{3.13}$$

We compare the performance of our approach with three baselines; *KeyGraph* approach described in [96], Semantic Expansion of *Hashtags* approach described in [78], and *EnBlogue* approach described in [4]. We also compare with our previous work (i.e., *LEED*) described in [107] in order to see how well slang word conversion and synonym expansion improve our results. Three baselines are described as follows:

- ***KeyGraph* approach** uses a graph of extracted terms where nodes are the terms and edges between the nodes are formed when those terms co-occur in a document. The researchers apply community analysis techniques to the graph adapted from social network theory to discovery events. The source code of this approach is provided in their project[9].

- ***Hashtags* approach** is a co-occurrence based method for identifying the semantic relationships

[9]http://keygraph.codeplex.com

among the *hashtags*. This method uses only the *hashtags* in *Twitter* messages to generate document vectors and applies a lexico-semantic expansion on *hashtags* to document vectors. For each *hashtag*, the researchers count the co-occurrences of *hashtags* with *non-hashtags* words in *Twitter* messages and generate co-occurrence vectors. The top three *hashtags* which have the highest cosine similarity scores are selected to expand in a document vector and then used in the clustering process. Set of *hashtags* is reported as an event. *Twitter* messages with a single *hashtag* are used in this approach.

- ***EnBlogue*** **approach** is the correlation between two *tags* which indicate an emerging topic. *Tag* is a simple annotation to a document explicitly given, or name entities extracted from the document. For *Twitter* messages, *tags* are *hashtags* and named-entities like people, organization and places found in the messages. *Seed tags* are selected based on popularity. For each *tag pair* that contains at least one *seed tag*, the researches keep track of their correlations. Next, they detect shift on related *tag pairs* to identify emerging events. In the final step, two *tag pairs* are merged in the same group if they co-exist in 80% of the messages.

TABLE 3.6: Detection results of LSED against baseline methods in Country level.

Method	**# of detected events**	**# of real-life events** (A)	**# of real-life local events** (B)	**# of distinct real-life events**	**# of distinct real-life local events**	**Precision** (B/A)	**Recall**
KeyGraph	54	23	20	15	12	0.870	12
Hashtags	949	23	21	17	13	0.913	13
EnBlogue	1024	220	190	121	82	0.864	82
LEED	151	121	117	95	90	0.967	90
LSED	136	112	109	94	90	**0.973**	**90**

According to the limitation of our collected data, we set the sliding window size to six hours and previous time slots to three blocks in order to compute emerging events. The value of these parameters depends on the user preference. We present the results of the experiments in Table 3.6.

It can be seen that our approach can effectively detect emerging hotspot events with a precision of 0.973 which is significantly larger than the baselines. In other words, the correlation between user location and event location can filter non real-world event clusters out of our system. Our approach

can also detect a larger number of real-world local events (90 distinct events) than the baselines. By applying slang conversion, synonym expansion and conceptual similarity of terms to provide a rich semantic context for measuring message similarity can reduce the number of duplicated real-world events and improve the results of the clustering method.

3.4 Summary

In this study, our results show that by considering the correlation between user location and event location can help to detect real-world events from micro-blog messages better than the three baseline approaches. The discussion of each approach is described as follows.

The researchers in *KeyGraph* approach used the traditional TFIDF for term weighting. However, the TFIDF is not designed to deal with a very short text and noisy data. As we can see in Table 3.3, the TFIDF is not performing well in this circumstance. Their approach is also sensitive to the parameter setting which affects the performance of the approach. The parameters consist of the minimum number of documents that contain each keyword, minimum number of co-occurring keywords in the same document, minimum cluster node size, and maximum cluster node size. Nodes and edges of graph are filtered out according to these parameters. According to our experiment, a lot of nodes and edges are removed because they are not meeting the conditions. In our experiment we use the default setting which is given for them.

Both *Hashtags* approach and *EnBlogue* approach rely on using *hashtags* in a document. *Hashtags* approach uses only documents that contain single *hashtags* while *EnBlogue* approach uses all documents. These two approaches suffer from the problem of ignoring all documents that do not contain a *hashtag*. *Hashtags* approach also ignores messages which contain more than one *hashtag* such as,

"Tropical Storm Debby spawns fatal tornado in Florida, drenches coast http://t.co/ZeI7bGjt ***#cnn #florida #hurricaine #debby****"*

According to our experiments, approximately 70 percent of messages do not contain any *hashtags*. Also, a lot of *hashtags* are not related to the real-world events such as *#autofollowback, #FF*

(FollowFriday), #birthdaywish, #iloveyousomuch and etc. According to the above reasons, *Hashtags* approach can only detect 23 real-world events while the rest of them are not.

In *EnBlogue* approach, they also combine *Name Entity Recognition (NER)* with the *hashtags*. *NER* is used for extracting entities like people, organizations and places in the documents. The extraction result relies on the service they used. Even though this approach can detect more real-world events, it still suffers from non-related event *hashtags* that can increase the number of non-real world event clusters. Moreover, this approach detects more duplicated events than our approach, such as *"the Tropical storm Debby"* event. The sample event clusters are included *"#debby, gulf, florida", "#debby, #gulfcoast, #tropicalstorm", "#debby, fla, gulf coast, florida", "florida, gulf of mexico"* and etc. The *tag pairs* are not grouped to the same event because their co-exist in less than 80 percent of the messages (according to their threshold parameter).

TABLE 3.7: Sample of top 5 events detected by *KeyGraph* approach on 25/06/2012

Detected events	**Description**
1) thunder,spawns,stalls,tropical, storm, tornadoes,threatened	Tropical Storm Debby stalls in Gulf of Mexico
2) park,police,office	Police officer shot dead at jazz concert in Denver park
3) truck,sell,right	City's red fire trucks to be transformed into billboards
4) township,vehicle,south,unknown	Vehicle accident, Hempfield and Manheim Township
5) alarm,calories,brush,burn,acres	Brush fire burns 50 acres on Kent Island

TABLE 3.8: Sample of top 5 events detected by *Hashtags* approach on 25/06/2012

Detected events	**Description**
1) #dead,#ifeelstupid,#oomf,#whydidiwakeup	not real-world event
2) #favouritemoviequote,#favouritemoviequotes,#hawt, #thingsidliketohear	not real-world event
3) #dead,#forreal,#oomf,#simperingbitch	not real-world event
4) #mybad,#sleepingdisorder,#weupallnight,#whatreallyturnmeon	not real-world event
5) #Debby,#debbie,#dc,#phillies	Tropical Storm Debby

For our approach, we choose to deal with entire messages, instead of single or pairs of words. Our approach is about detecting emerging stories not emerging words because mapping clusters of words to real-world events is a very difficult and subjective task. For example, the group of keywords "Obama, Romney" can be the debate event or the poll result in many States or a discussion of the election campaign or even, any topic related to "Obama" and "Romney" in the US election 2012, which are hard to interpret.

TABLE 3.9: Sample of top 5 events detected by *EnBlogue* approach on 25/06/2012

Detected events	Description
1) #debby,florida,gulf of mexico, hurricaine,tropical storm debby	Tropical Storm Debby
2) detroit free press,#obama,#romney, michigan	Polls about Obama and Romney in Michigan (not real world event)
3) #debby,#gulfcoast,tropicalstorm	Tropical Storm Debby (related to event 1)
4) #debby,fla,gulf coast,florida	Tropical Storm Debby (related to event 1)
5) firefighters,philly,fireflames,housefire, onealarm,police,riversident,smokedout	Riverside NJ fire - Philadelphia

Moreover, our approach can achieve high *Precision* score because it can filter out non-real world event clusters. Most of non-real world event clusters do not contain event locations. For example, the cluster which contains *hashtag "#FF" (i.e., FollowFriday)* is emerging only on Friday and it will be removed from our approach because it does not contain the event location mentioned in the cluster. However, using slang conversion failed in resolving ambiguity such as the word *"cm"*, it can be *"come"* or *"centimetre"*. For illustration purposes, we present examples of detected events on *25/06/2012* of the baseline approaches in Tables 3.7, 3.8 and 3.9. Table 3.10 presents our top five detected events during 21 June 2012 to 25 June 2012. Table 3.11 presents our top five detected events during 26 June 2012 to 27 June 2012. However, only one event is detected on 23/06/2012 and three events are detected on 24/06/2012 from our approach.

In this chapter, an approach namely *LSED*, to automatically detect emerging hotspot events with location sensitivity over micro-blogs is developed. The goal of our approach is to effectively detect emerging hotspot events by utilizing real-time micro-blog messages and location information (i.e., user location and event location). Our contributions are summarized as follows:

- An effective method to detect the emerging hotspot events is proposed.
- An approach to correlate user location with event location in order to establish a strong correlation between them is proposed to identify hotspot events.
- An algorithm is designed for slang conversion, synonym expansion and conceptual similarity to provide a rich semantic context for measuring message similarity to improve clustering results.

TABLE 3.10: Sample of top 5 events detected by the *LSED* approach between 21-25 June 2012

Detected events	Description
Date: 21/6/2012	
1) chief, fire, martin, polic, shoot, trayvon	Police chief fired over Trayvon Martin shooting
2) hold, rebel, syrian, talk, weapon, ya	Syrian rebels hold talks with US over weapons
3) american, fisher, mark, photograph	not real-world event
4) affect, ariza, emeka, okafor, plan, trade, trevor, washington	not real-world event
5) flight, kid, morgantown, take	Kids take flight over Morgantown
Date: 22/6/2012	
1) 26, afghan, die, fight, forc, hotel, kabul, taliban, time, york	26 die as Afghan Forces fight Taliban at Hotel near Kabul
2) 2012, act, birmingham, fest, goon, jamz	Jamz Fest In Birmingham
3) believ, coyot, fairfield, pet, respons, seri	Coyotes believed responsible for series of missing pets in Fairfield
4) 34th, amber, brunmeier, kalamazoo, klassic	34th Kalamazoo Klassic
5) 2012, champion, heat, miami, nba	NBA match
Date: 23/6/2012	
1) cheney, dc, famili, legal, marri, rejoic	The Cheney family rejoices as Mary is finally able to legally marry in DC
Date: 24/6/2012	
1) avoid, destini, fontain, jean, meet, person	not real-world event
2) advisori, atlant, debbi, storm, tropic	announce to prepare for Tropical storm Debby in Atlanta
3) citi, detroit, feel, hard, love, straight, world	not real-world event
Date: 25/6/2012	
1) behold, dalla, debbi, editor, storm, tropic	A tropical storm named Debbie, headed for Dallas
2) debbi, florida, gulf, move,slowli, soak, storm, tropic	Moving Slowly in Gulf,Tropical Storm Debby Soaks Florida (Related to event 1)
3) accid, buse, circl, injuri, involv, metro, report, washington	Accident Washington Circle 2 Metro buses involved.
4) doppler, indic, lightn, nyc, rotat, storm, thunder, trenton	Storm from Trenton to NYC
5) accid, close, delay, lane, mm133, potenti, stafford	Accident at MM133 in Stafford Co

- An effective evaluation for event detection on a real-world *Twitter* dataset with different granularities of locations is performed.

Our experiments are performed against three baseline approaches. The results show that our approach is effective in detecting emerging hotspot events. However, further improvements are needed with respect to the use of *Gazetteer* in our approach in the granularity of locations and the speed of

TABLE 3.11: Sample of top 5 events detected by the *LSED* approach between 26-27 June 2012

Detected events	Description
Date: 26/6/2012	
1) dalla, debbi, prayer, storm, tropic	prayer to tropical Storm Debby
2) accid, avenu, brentwood, inbound, lane	Accident in Brentwood, New York
3) beach, debbi, englewood, mexico, move, slow, storm	Tropical storm Debby move slowly to Englewood, Florida
4) accid, close, delay, fairfax, lane, mm174	Accident at MM174 lane in Fairfax
5) alarm, condo, fire, live, look, reston	Apartment fire in Reston
Date: 27/6/2012	
1) colorado, feroci, fire, flee, spring	Thousands flee ferocious fire in Colorado Springs
2) bomb, cargo, nuclear, shipment, terrorist	The cargo containers arriving on ships from foreign ports offer terrorists
3) 1200, alarm, baltimor, block, fire, street, tune, west	2-alarm fire in the 1200 block of West Baltimore Street
4) colorado, debbi, extrem, fan, fire, florida	A tropical storm Debby in Colorado
5) accid, clear, mm78, richmond	Accident at MM78 in Richmond

processing. In future work, other on-line clustering methods will be compared on the effectiveness of grouping the messages into events. The algorithms on *Geospatial Named Entities Recognition* will be further studied to improve location extraction from social network messages.

Chapter 4

Sub-Event Detection and Sentiment Analysis in Social Networks

4.1 Problems and Challenges

Social networks are widely used by all kinds of people to express their opinions. People express themselves spontaneously with respect to the social events in their social networks. For a long-running event like a nation-wide election which usually has fixed start and end times, users may want to monitor sub-events. We define sub-events in this work as follows:

Definition 4.1. *"Sub-events" is hierarchically nested events that break down an event into more refined parts such as the debate or campaign-launch speech.*

Alternatively, policy-makers may want to know the feelings of users during the course of an election. The new research in computer science, sociology and political science shows that data extracted from social media platforms yield accurate measurements of public opinion. It turns out that what people say on *Twitter* is a very good indicator of how they would vote in an election [77, 105, 93, 64]. In this chapter, we present an approach by incorporating sub-event detection and sentiment analysis to analyse as well as visualise political preferences revealed by those social network users. To evaluate our approach, we utilize our approach to predict the election results at a state as well as a national level, as a case study.

Recently, extensive research has been done on social networks in election prediction [77, 105, 93, 64]. O'Connor et al. in [77] present the feasibility of using *Twitter* data as a substitute and supplement for traditional polls. Subjectivity lexicon is used to determine opinion scores (i.e., positive and negative scores) for each message in the dataset. Then, the authors computed a sentiment score. Consumer confidence and political opinion are analysed and found to be correlated with sentiment word frequencies in *Twitter* data. However, they do not describe any prediction method. Tumasjan et al. in [105] examine whether *Twitter* can be seen as a valid real-time indicator of political sentiment. The authors also found that the mere number of messages reflects the election result and comes close to traditional election polls. Sang et al. in [93] analyse *Twitter* data regarding the 2011 Dutch Senate elections. The authors presented that improving the quality of the document collection and performing sentiment analysis can improve performance of the prediction. However, the authors need to manually annotate political messages to compute sentiment weight and only the first message of every user is taken into account. In addition, the method relies on polling data to correct for demographic bias. Makazhanov et al. in [64] propose political preference prediction models based on a variety of contextual and behavioural features. The authors extract all interactions of the candidates, group them on a per-party basis, and build a feature vector for each group. Both a decision tree-based J48 and Logistic regression classifiers are utilized for each party. However, this method needs labelling of training examples for each user. The labelling of training set based on a set of users whose political preferences are known based on the explicit statements (e.g., *"I voted XXX today!"*) made on the Election Day or soon after. Moreover, it does not predict the election outcomes.

However, there are several works presented the problems on election prediction using *Twitter* data. Jungherr et al. in [44] present that a lack of well-grounded rules for data collection and the choice of parties and the correct period in particular can cause the problems. Metaxas et al. in [70] conclude that *Twitter* data is only slightly better than chance when predicting elections. However, the authors described three necessary standards for predicting elections using *Twitter* data: (1) it should be a clearly defined algorithm, (2) it should take into account the demographic differences between *Twitter* and the actual population, and (3) black-box methods should be avoided. Gayo-Avello has criticize several flaws in [26]. For example, there is not a commonly accepted way of counting votes in *Twitter*. Sentiment analysis is applied as a black-box and demographics are neglected. Nevertheless, the author

has outlined some of the research lines for future works in this topic. For example, researches need to clearly define which are a vote and the ground truth; sentiment analysis is a core task and researches should acknowledge demographic bias.

Existing studies in predicting election outcomes from social networks have focused on counting of preferences or sentiment analysis on a party or candidate. They neglect the fact that the voters' attitudes and opinions of people may be different depending on specific political topics and in different geographic areas. Moreover, the same voters participating in different discussions may have different political preferences. In this chapter, we are interested in predicting the result of elections from micro-blog data by incorporating sub-event detection and sentiment analysis to detect their political preferences and predict the election results at a state as well as a national level.

The main contributions of this work are as follows.

- We present an approach to forecast the vote of a sample user based on the analysis of his/her micro-blog messages and count the votes of users to predict the election results.

- Sub-event detection and sentiment analysis are incorporated to predict the vote of users as different level of sub-events user engaged in the discussions will affect the prediction results.

- We evaluate our proposed approach with a real-world *Twitter* data posted by Australia-based users during the 2013 Australian federal election.

4.2 Sub-Event Detection and Sentiment Analysis Approach

In order to provide a complete coverage of sub-event detection and sentiment analysis in social networks, we proposed our approach which has four stages as presented in Figure 4.1. Firstly, the pre-processing is performed to remove irrelevant data from the dataset. Secondly, we conduct a clustering approach to automatically group the messages into sub-events. Thirdly, we propose a lexicon-based approach to detect users' opinions for specific entities. Finally, we develop a visualization model for representing sub-events and users' opinions.

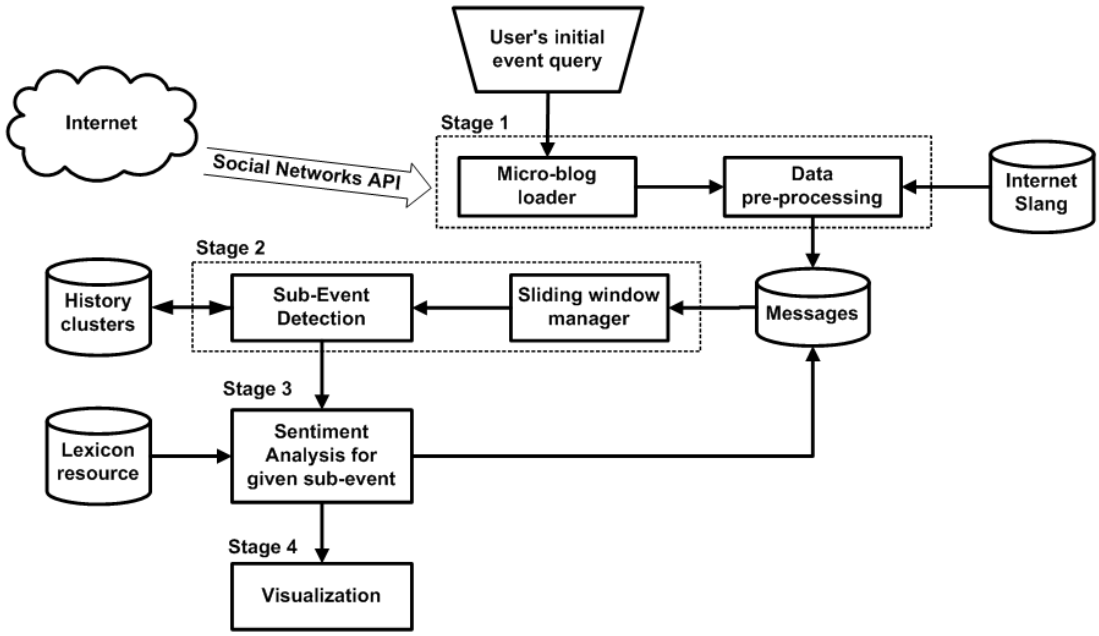


FIGURE 4.1: Architecture of Sub-Event Detection and Sentiment Analysis (SED-SA) system.

4.2.1 Sub-Event Detection for a Particular Event

The notion of event detection was proposed in our recent work [108] for location-based hotspot emerging events. However, the problem that we address in this work is how to group a set of micro-blog messages into a cluster (or sub-event) for a particular longer-running event (i.e., an election). The user defines an event by specifying a keyword query. For example, search keywords such as *"election", "Kevin Rudd", "Tony Abbott", "#ausvote" and "#auspol"* are used to collect the data of the 2013 Australian federal election. In the following, we brief the techniques for sub-event detection.

It has three steps as we are not consider the emergence of event. Firstly, the pre-processing was designed to ignore common words that carry less important meaning than keywords and to remove ir-relevant data e.g., *re-tweet* keyword, web address and message-mentioned username. Slang word and extensions like "booooored" are replaced by proper English words. The stop words are removed and all words are stemmed by using *Lucene 3.1.0 Java API*[1]. Message location identification is conducted in order to understand users' opinions in particular areas. We firstly extract message location from the *geo-tagged* (latitude/longitude) information. If *geo-tagged* information is not available we extract user location in the user profile to query the Australia *Gazetteer* database for acquiring the location's

[1] http://lucene.apache.org

address. Then, if neither of them is available we set user location equal to "Australia".

Secondly, for clustering step, we consider a set of messages where each message is associated with a sub-event. With the number of sub-events being unknown in advance, we applied event detection using hierarchical clustering from our previous work [108] with some modifications. We use a sliding window to divide the messages. The size of the sliding window is defined in time intervals (i.e., one day for our experiment). According to our experiment, the clustering method performs well when using the augmented normalized term frequency and cosine similarity function. The cosine similarity function is used to calculate the similarity between the existing cluster and the new message. Every message is compared with all previous cluster's centroids. The algorithm creates a new cluster for the message if there is no cluster whose similarity to the message is greater than the threshold (α). In order to find the most suitable value for the threshold, we conducted the clustering experiments with different threshold values. Our tests show that when $\alpha = 0.30$ it renders the best performance. The mean is used to represent the centroid of the cluster, which trades memory use for speed of clustering.

Finally, after the clustering is performed, all clusters cannot be assigned as event clusters because they can be private conversations, advertisements or others. A cluster can be considered as sub-event if there is strong correlation between the event location (i.e., location mentioned in the messages) and the user location. For event location identification, we find all terms or phrases which reference geographic location (e.g., country, state and city) from message contents. We simply extract the message-mentioned locations via *Named Entity Recognition (NER)*. We use the *Stanford Named Entity Recognizer* [45] to identify locations within the messages. We also use the *Part-of-Speech Tagging* for *Twitter* which is introduced in [27] to extract proper nouns. We use an extracted terms query into the *Gazetteer* database to obtain candidate locations of the event. We find the most probable location of the event using the frequency of each location in the cluster. The location which has the highest frequency is assigned as the event location. In order to understand what the sub-event cluster is about, we find the set of keywords to represent the sub-event topic. To extract the set of co-occurring keywords, firstly we create a directed, edge-weighted graph. We adopt the smoothed correlation weight function, to calculate the semantic correlation weight between terms. We identify the sub-event topic by extracting the *Strongly Connected Components (SCCs)* from the graph. The details of our algorithm are presented in Chapter 3.

4.2.2 Political Sentiment Analysis

In general, opinions can be expressed about anything, such as a product, service, person, topic or event and by any person or organization. Entity is used to denote the opinion target. For example, the targets/entities of messages likes *"As much as you dislike XXX please Australia...Hate YYY more! I beg you"* are *"XXX"* and *"YYY"*. Formally, we have the following:

Definition 4.2. *An "entity" e is a person or candidate in the 2013 Australian Federal Election. It is represented by a set of referred candidate name $A = \{a_1, a_2, ..., a_n\}$.*

An opinion is simply a positive or negative sentiment, attitude, emotion or appraisal about an entity or an aspect of the entity from an user. Positive, negative and neutral are called opinion orientations (also called sentiment orientations, semantic orientations, or polarities) [61]. An opinion in our work is defined as follows:

Definition 4.3. *An "opinion" is a quintuple, $(e_i, u_j, tk, c_l, o_{ijkl})$, where e_i is the name of an entity, u_j is the user, t_k is the time when the opinion is expressed by u_j, c_l is the sub-event cluster that user u_j had engaged in, and o_{ijkl} is the orientation of the opinion about entity e_i. The opinion orientation o_{ijkl} can be positive, nagative or neutral.*

Sentiment analysis can be a supervised approach or an unsupervised approach or a combination of the two. In the supervised approach, the process of labelling training datasets requires considerable time and effort. Collecting training datasets for all application domains is very time consuming and difficult. In this work, we focus on a lexicon-based approach to perform sentiment classification. However, spotting the target/entity in a micro-blog message is not the focus of this work. Our method has two steps. First, an opinion lexicon is constructed and then, the opinion is classified, based on a statistical calculation.

For sentiment analysis, the pre-processing is conducted. We performed the part of speech (*POS*) processing from the original messages. We use *Twitter NLP and Part-of-Speech Tagging* proposed by Gimpel et al. in [27] for tagging the messages. Moreover, the *emoticons* are extracted from the messages. Finally, all messages after being tagged are stored in the database.

1) Opinion Lexicon: We used the lexicon dictionary which was introduced in [36]. It consists of 4,783 negative and 2,006 positive, distinct words. However, micro-blog messages are informally

TABLE 4.1: Natural language rules for phrase detection.

Rule	Example
Adverb + Adjective	not good, very sad
Comparative Adverb + Adjective	more offensive, more sincere
Adverb + Verb	not vote, never truth
Intensifier/Diminishes + Adverb	really good, slightly nervous
Modals Verb + Verb	can't promise, can't believe

written and often contain slang words and abbreviations. The traditional lexicon dictionary does not cover opinion words in micro-blogs. In order to expand the lexicon dictionary, we manually annotated the Internet slang dictionary, downloaded from http://www.noslang.com, into 262 positive and 903 negative slang words. *Emoticons*[2] are also grouped into happy and unhappy facial expressions.

2) Lexicon-based Algorithm: Our algorithm assigns the messages into positive, negative and neutral classes. Given a message, the tasks are divided into three steps: word-level sentiment, aspect-level sentiment and sarcasm identification.

Word-level sentiment: This step aims to mark all opinion words or phrases in the message. Each positive word is assigned an opinion score of $+1$ while each negative word is assigned the score of -1. We extracted adjectives, adverbs, verbs, nouns, interjections and *hashtags* to assign the opinion score. Also, the happy emoticon is assigned the opinion score of +1 and vice versa. In order to detect a phrase, we applied natural language rules which are shown in Table 4.1.

In this step, it is important to deal with complex linguistic constructions, such as negation, intensification, diminishes and modality because of their effect on the emotional meaning of the text. Negation and modality are computed in the same way. We defined the rules for negation and intensification as follows. For negation (e.g., "no", "not" and "never"), there are three cases to compute an opinion score (OS) of a given phrase.

(1) Negation + Neg. e.g., "not bad"; $OS = +1$

(2) Negation + Pos. e.g., "not good"; $OS = -1$

(3) Negation + Neu. e.g., "not work"; $OS = -1$

Intensifiers (e.g., "very", "really" and "extremely") increase the semantic intensity of a neighbouring

[2]http://en.wikipedia.org/wiki/List_of_emoticons

lexical item, whereas diminishes (e.g., "quite", "less", "slightly") decrease it. The opinion score of a phrase is computed as follows.

(1) Intensifier + Neg. e.g., "very bad"; $OS = -1.5$

(2) Intensifier + Pos. e.g., "very good"; $OS = 1.5$

(3) Diminishes + Neg. e.g., "slightly mad"; $OS = -0.5$

(4) Diminishes + Pos. e.g., "quite good"; $OS = 0.5$

Aspect-level sentiment: In this step we aim to compute the opinion orientation for each aspect/target. For the message likes *"As much as you dislike XXX please Australia...Hate YYY more! I beg you"*, we want to extract a pair of opinion word and the aspect such as {*"dislike"* and *"XXX"*} and {*"hate"* and *"YYY"*} then we can calculate the aspect-level score. We applied an opinion aggregation function to assign the final opinion orientation for each aspect in the message. Each aspect has many names that refer to it, even within the same message and clearly, across messages. For example, {*"Tony Abbott", "Abbott" and "TonyAbbottMHR"*} refer to the same person who is one of the candidates of the 2013 Australian federal election. As extracting the aspect/target in micro-blog messages is not the focus of this work, we simply set the aspects of our experiments to two sets of keywords as follows:

A_1 = {*"Tony Abbott","Abbott","TonyAbbottMHR"*},

A_2 = {*"Kevin Rudd", "Rudd", "KRudd", "KRuddPM"*}

Every word opinion score is computed related to its distance to the aspect. The number of words between the current word and the aspect (i.e., the matched keywords in the aspect keyword set) is assigned as the distance of the current word to the aspect. The aspect-level score is computed as:

$$asp\_score(m, A) = \sum_{w_i \in m} \frac{opinion\_score_{w_i}}{min(distance(w_i, a)), a \in A} \tag{4.1}$$

where m is the message, A is the set of aspect keywords, w_i is the word in the messages m and a is the aspect keyword in A. The aspect sentiment is positive if $asp\_score(m, A) > 0$, and is negative when $asp\_score(m, A) < 0$. Otherwise, the aspect sentiment is neutral.

TABLE 4.2: The statistical information of sarcasm messages.

List	Kevin Rudd	Tony Abbott
No. of messages	1,481	3,254
No. of users	959	1,737
No. of users who posted sarcastic messages	48	114
% of users who posted negative sarcasm	100.00%	100.00%
% of users who have the same opinions in every message for a given topic/event	89.58%	92.98%
% of users who have both positive and negative messages for a given topic/event	10.42%	7.02%

Sarcasm identification: In addition, micro-blog messages also contain extensive use of irony and sarcasm, which are particularly difficult for a machine to detect [28]. Sarcasm transforms the polarity of the message into its opposite. Negative sarcasm is a message that sounds positive but is intended to convey a negative attitude. Positive sarcasm is a message that sounds negative but is apparently intended to be understood as positive. Watching people's faces while they talk is a good way to pick up on sarcasm. However, it is very difficult to detect sarcasm in writing due to lack of intonation and facial expressions.

In order to understand the sarcastic messages in micro-blogs, we conducted statistical studies. We manually labelled 5,735 messages sent by users around Australia related to one sub-event (i.e., the first debate of the 2013 Australian federal election between *Kevin Rudd* and *Tony Abbott* on 11 August 2013 from 6pm to 9pm). There are 1,481 and 3,254 messages which discussed *Kevin Rudd* and *Tony Abbott*, respectively. The messages are annotated with the polarity being positive, negative or neutral and are also marked as sarcastic messages where applicable. The statistical information for sarcasm is shown in Table 4.2.

As we can see from Table 4.2, most users hold negative views on sarcastic messages. Our interest in this task is to mark off whether a message is intended to be sarcastic and assign the polarity of the message. Considering a single message, it is very difficult to classify sarcasm, even for humans. In general, a message like *"XXX: Road is the future of transport! Brilliant."* will be considered as a positive opinion; however, some people in developed countries might think this is a sarcastic message as they have too many roads now. Therefore, the message itself cannot be effective to predict sarcastic message. The previously messaged opinions of the author may help to classify whether the current

Algorithm 7: $LexiconClassification(m, asp, prevMsgs)$

Input: m: the current message, asp: the aspect/target, $prevMsgs$: the previous messages in the same sub-event of the message m

Output: pol : the polarity of the message m

```
asp_score = 0;
pos = getPartOfSpeech(m);
segments = extractWordAndPhrase(pos);
for s ∈ segments do
    if (s is a word) then
        r = getWordLevelSentiment(s);
    else
        r = getPhraseLevelSentiment(s);
    dist = distance(m, s, asp);
    asp_score+ = r/dist;
if (asp_score > 0) then
    pol = 1;
else if (asp_score < 0) then
    pol = −1;
else
    pol = 0;
if (pol <> 0) then
    pol = isSarcasm(m, pol, prevMsgs);
return pol;
```

message tends to be sarcastic or not. However, some people may have different opinions on different topics/sub-events. Based on our observation on sarcasm in micro-blogs we found that most of the micro-blog users have only one opinion on a specific topic or event (89.58% and 92.98% of messages related to *Kevin Rudd* and *Tony Abbott* respectively). The message examples of a given user on the first debate event are shown below:

M1: *"I'm already **sick** of XXXX's slogans and **lies**. #YouDecide9"*

M2: *"Mr XXXX is **lying**. The GSt can change with an act of parliament."*

M3: *"@Aussieboyd I agree. It is a load of **shit**. Mr XXXX is a **bad** man."*

M4: *"Hearing XXXX talk about "the campaigns" is too **funny**."*

The message $M4$ sounds positive but is intended to convey a negative feeling according to his/her previous messages. Therefore, a reasonable ways to classify sarcastic messages are to consider a specific facial expression (i.e., *emoticon* expression) and to compare them with the author's previous

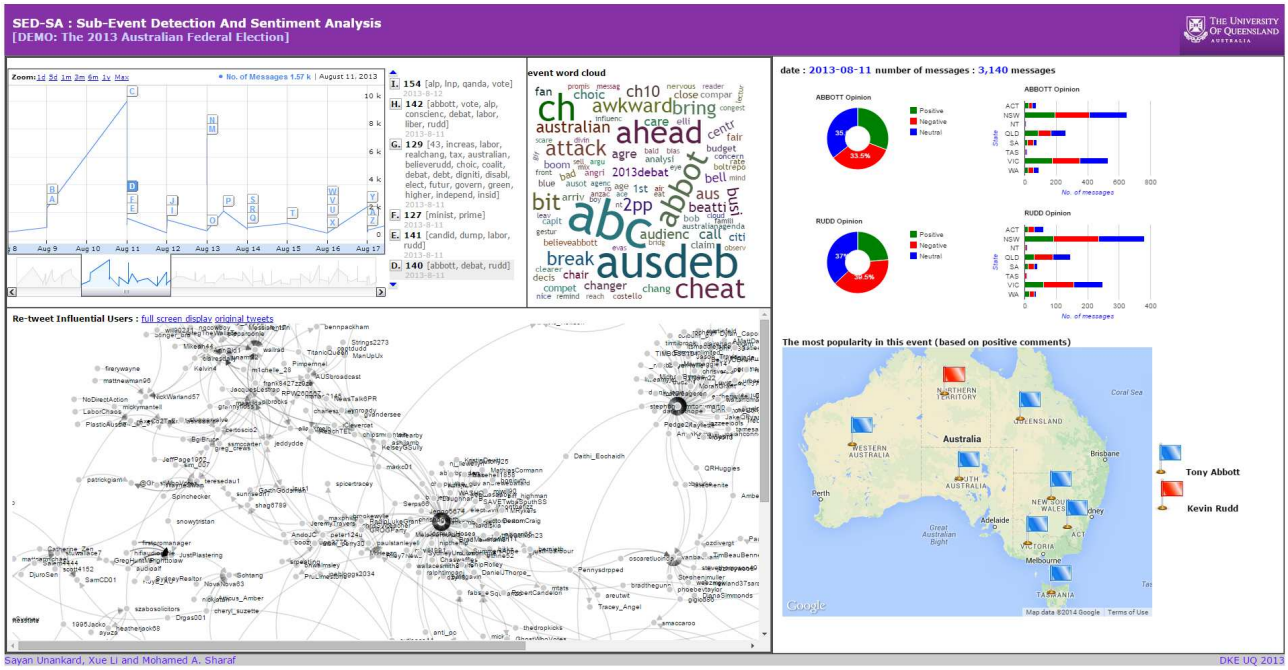


FIGURE 4.2: A dashboard to display sub-event and sentiment of two specific candidates.

messages in the same topic or sub-event. To address this issue, the *emoticon* expression will be compared with the message polarity. All messages accompanied with an *emoticon* are computed as follows.

(1) Pos. message + Neg. *emoticon*; $polarity = -1$

(2) Neg. message + Pos. *emoticon*; $polarity = +1$

If there are no *emoticons* in the message, we compare the aspect opinion score of the current message with the previous messages of the author in the same sub-event and within the same interval of time (i.e., the size of the sliding window). If the current message opinion differed from the overall opinions of previous messages in the same sub-event (i.e., greater than 90%), we change the aspect opinion score of the current message to the same as that for the previous message's opinion. However, if the opinions of the previous messages are divergent, the current message opinion is not changed because it is surmised that this author tends to have different opinions on the same sub-event. The algorithm is shown in Algorithm 7.

For usability and understanding issues of visualizing the model, we designed a dashboard to display sub-event and sentiment of two specific candidates. Sub-events are presented via *Annotated Time*

Line Chart[3], an interactive time series line chart with optional annotations provided by *Google API* as shown in Figure 4.2 (top-left) for each day (represented by letters A to Z). For given event, the sub-event is represented by word cloud as shown in the middle of the dashboard. The importance of each word is shown with font size. Figure 4.2 (right) displays how people feel about specific opinion targets for a given sub-event. In this work, we set the opinion targets to two candidates in the 2013 Australian federal election (i.e., *Kevin Rudd* and *Tony Abbott*). The sentiment orientations for each aspect/target in a region are illustrated via a Bar chart. Figure 4.2 (bottom-left) displays network graph to represent re-tweet influential users.

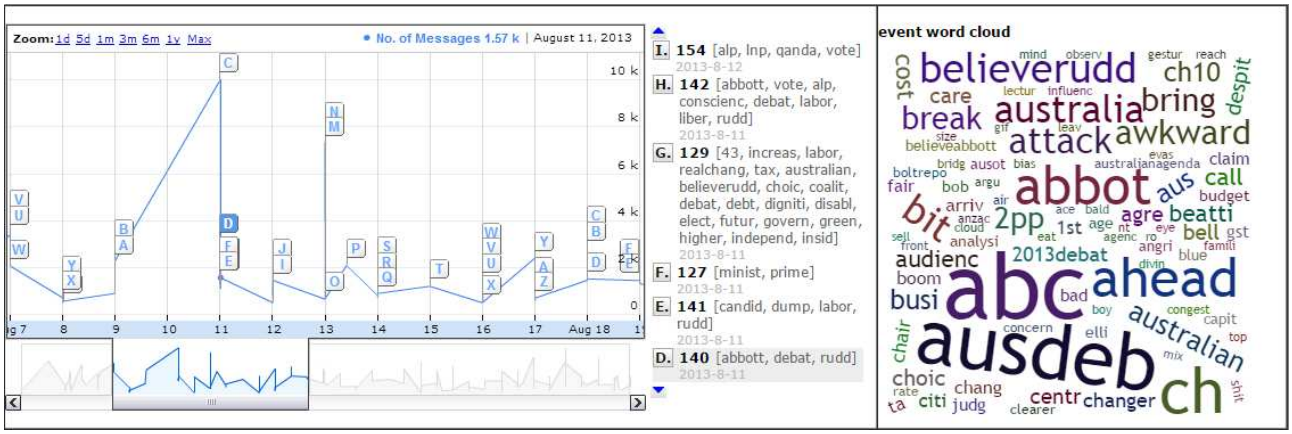


FIGURE 4.3: Annotated time line chart and *WordCloud* to display sub-events.

Figure 4.3 illustrates all sub-events and selected sub-event topic. In order to display sub-events, annotated time line chart is used. These time line illustrates the big picture of events and help readers understand the major events that happened during that time. User can change the range slider of date. When user select a particular event, *WordCloud*[4] of the selected event will be displayed. *WordCloud* displays the most frequently used words for this event topic. The size of word indicates the frequency of word over the selected event topic.

Figure 4.4 shows users' opinions for a given sub-event topic. Pie chart illustrates three types of users' opinions including positive (green), negative (red), and blue (neutral) feeling. The bar chart is used to represent users' opinions in each state. The candidate who has the most positive comments is displayed on the *Google* map for each state.

[3]https://google-developers.appspot.com/chart/interactive/docs/gallery/annotatedtimeline
[4]https://github.com/timdream/wordcloud2.js/blob/master/API.md

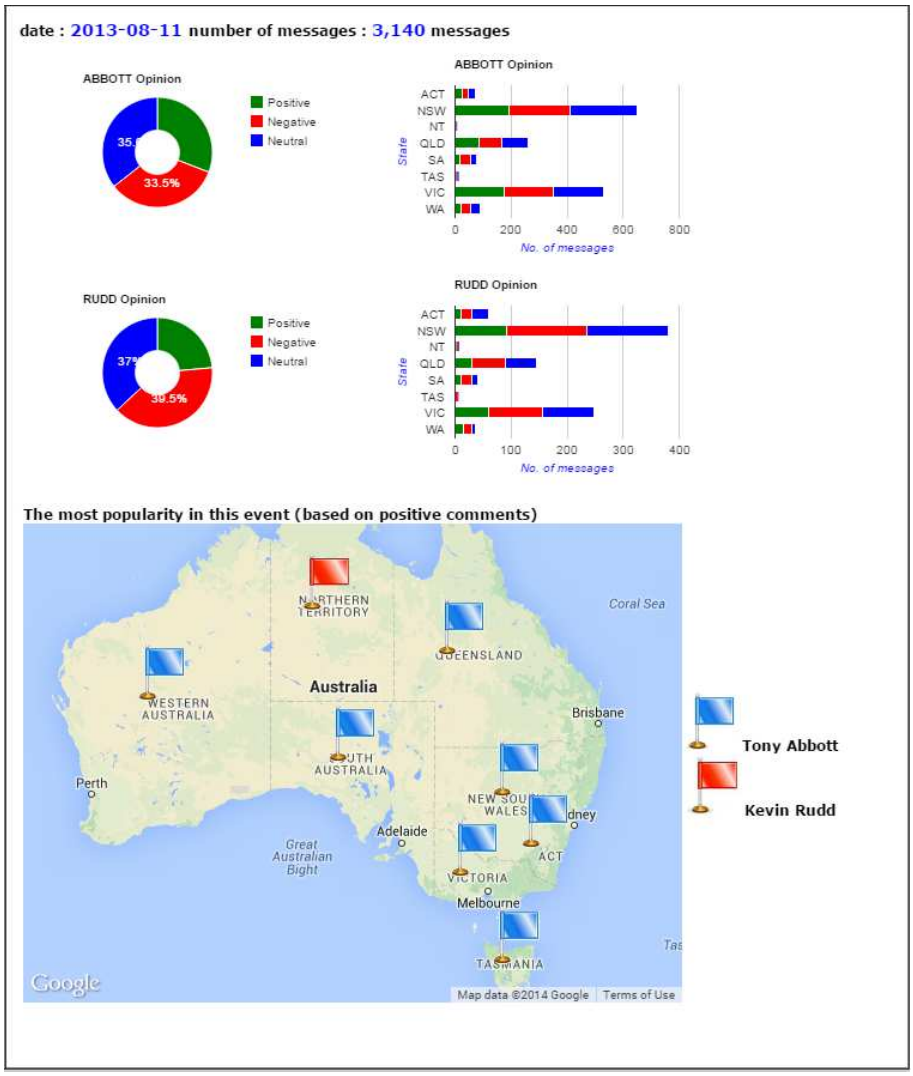


FIGURE 4.4: Visualizations for users' opinions of two specific candidates.

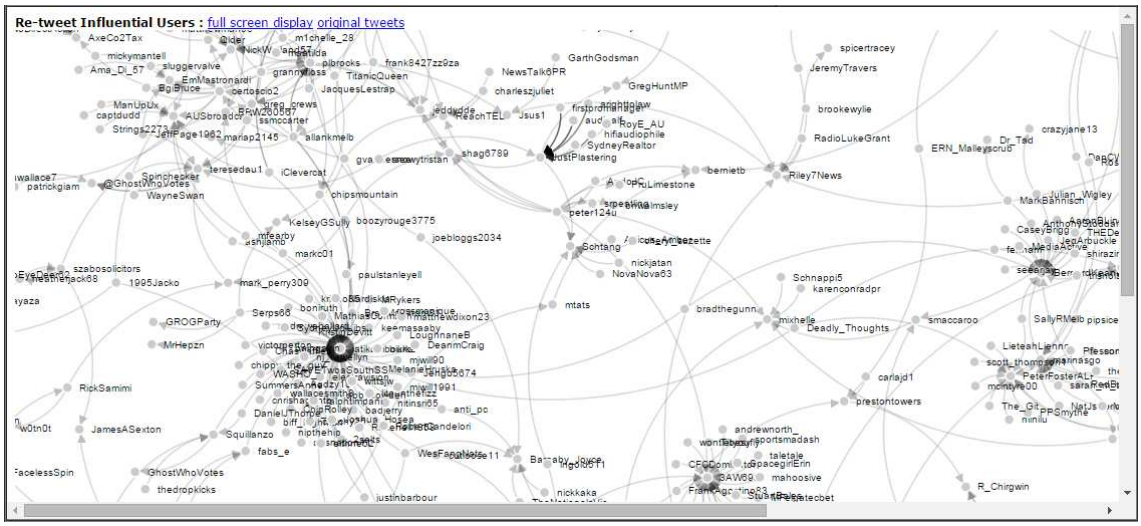
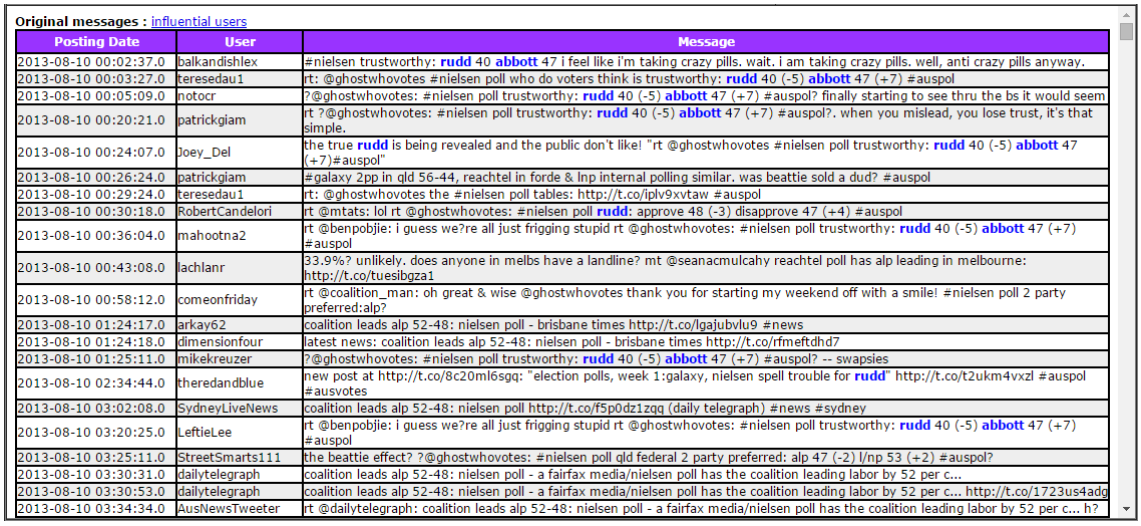


FIGURE 4.5: A network graph to display *re-tweet* influential users for a given sub-event.



Original messages : influential users

Posting Date	User	Message
2013-08-10 00:02:37.0	balkandishlex	#nielsen trustworthy: **rudd** 40 **abbott** 47 i feel like i'm taking crazy pills. wait. i am taking crazy pills. well, anti crazy pills anyway.
2013-08-10 00:03:27.0	teresedau1	rt: @ghostwhovotes #nielsen poll who do voters think is trustworthy: **rudd** 40 (-5) **abbott** 47 (+7) #auspol
2013-08-10 00:05:09.0	notocr	?@ghostwhovotes: #nielsen poll trustworthy: **rudd** 40 (-5) **abbott** 47 (+7) #auspol? finally starting to see thru the bs it would seem
2013-08-10 00:20:21.0	patrickgiam	rt ?@ghostwhovotes: #nielsen poll trustworthy: **rudd** 40 (-5) **abbott** 47 (+7) #auspol?. when you mislead, you lose trust, it's that simple.
2013-08-10 00:24:07.0	Joey_Del	the true **rudd** is being revealed and the public don't like! "rt @ghostwhovotes #nielsen poll trustworthy: **rudd** 40 (-5) **abbott** 47 (+7)#auspol"
2013-08-10 00:26:24.0	patrickgiam	#galaxy 2pp in qld 56-44, reachtel in forde & lnp internal polling similar. was beattie sold a dud? #auspol
2013-08-10 00:29:24.0	teresedau1	rt: @ghostwhovotes the #nielsen poll tables: http://t.co/iplv9xvtaw #auspol
2013-08-10 00:30:18.0	RobertCandelori	rt @mtats: lol rt @ghostwhovotes: #nielsen poll **rudd**: approve 48 (-3) disapprove 47 (+4) #auspol
2013-08-10 00:36:04.0	mahootna2	rt @benpobjie: i guess we?re all just frigging stupid rt @ghostwhovotes: #nielsen poll trustworthy: **rudd** 40 (-5) **abbott** 47 (+7) #auspol
2013-08-10 00:43:08.0	lachlanr	33.9%? unlikely. does anyone in melbs have a landline? mt @seanacmulcahy reachtel poll has alp leading in melbourne: http://t.co/tuesibgza1
2013-08-10 00:58:12.0	comeonfriday	rt @coalition_man: oh great & wise @ghostwhovotes thank you for starting my weekend off with a smile! #nielsen poll 2 party preferred:alp?
2013-08-10 01:24:17.0	arkay62	coalition leads alp 52-48: nielsen poll - brisbane times http://t.co/lgajubvlu9 #news
2013-08-10 01:24:18.0	dimensionfour	latest news: coalition leads alp 52-48: nielsen poll - brisbane times http://t.co/rfmeftdhd7
2013-08-10 01:25:11.0	mikekreuzer	?@ghostwhovotes: #nielsen poll trustworthy: **rudd** 40 (-5) **abbott** 47 (+7) #auspol? -- swapsies
2013-08-10 02:34:44.0	theredandblue	new post at http://t.co/8c20ml6sgq: "election polls, week 1:galaxy, nielsen spell trouble for **rudd**" http://t.co/t2ukm4vxzl #auspol #ausvotes
2013-08-10 03:02:08.0	SydneyLiveNews	coalition leads alp 52-48: nielsen poll http://t.co/f5p0dz1zqq (daily telegraph) #news #sydney
2013-08-10 03:20:25.0	LeftieLee	rt @benpobjie: i guess we?re all just frigging stupid rt @ghostwhovotes: #nielsen poll trustworthy: **rudd** 40 (-5) **abbott** 47 (+7) #auspol
2013-08-10 03:25:11.0	StreetSmarts111	the beattie effect? ?@ghostwhovotes: #nielsen poll qld federal 2 party preferred: alp 47 (-2) l/np 53 (+2) #auspol?
2013-08-10 03:30:31.0	dailytelegraph	coalition leads alp 52-48: nielsen poll - a fairfax media/nielsen poll has the coalition leading labor by 52 per c...
2013-08-10 03:30:53.0	dailytelegraph	coalition leads alp 52-48: nielsen poll - a fairfax media/nielsen poll has the coalition leading labor by 52 per c... http://t.co/1723us4adg
2013-08-10 03:34:34.0	AusNewsTweeter	rt @dailytelegraph: coalition leads alp 52-48: nielsen poll - a fairfax media/nielsen poll has the coalition leading labor by 52 per c... h?

FIGURE 4.6: The original *Twitter* messages for a given sub-event.

Re-tweet influential users are displayed in Figure 4.5. We used Force-Directed Graph[5] to create re-tweet network graph. User A *re-tweets* or re-posts a user B's message is represented by $A- > B$. The original messages are shown in Figure 4.6 for a selected sub-event including posting time, username and message content.

4.3 Election Prediction Model

In order to predict the election results, we learn from the professional pollsters. Our prediction model can be divided into two parts; sampling process and user's vote prediction. The messages since announce Election day (i.e., 4 August 2013) until the day before Election day (i.e., 6 September 2013) were used for predicting the results. Also, we decided to predict the two-party-preferred vote as in Australian politics the candidates will be from the two major parties.

4.3.1 Sampling Process

Since no one can be sure that who will actually vote, the prediction can be approximated by sampling those who will likely to vote. The most important aspect of correct prediction is the selection of a representative. We need to decide who is a particular sample of our prediction and how many people we need to predict. Almost all surveys rely on sampling. This work analyses a sample of *Twitter* users

[5]http://d3js.org/

in Australia. A user account which has *username* contains the words *"news"* and *"TV"* is removed (e.g., "abcnews", "abctv" and etc.) as it is news media account. We compute our sample size by using Cochran's sample size formula [17]. We want to estimate sample size (ss) with 95% confidence and the margin of error no larger than 3%. The formulas used in our sample size calculator are shown as follows:

$$n = \frac{Z^2 p(1-p)}{e^2}, \quad ss = \frac{n}{1 + (n-1)/P} \tag{4.2}$$

where Z is $Z-score$ corresponds to confidence level ($Z = 1.96$ for confidence level 95%), p is the maximum possible proportion (50% is the most conservative assumption), e is the acceptable margin of error (i.e., the amount of error that you can tolerate) and P is the population size. The minimum sample size (ss) for our experiments is 1,067 people. We randomly select the sample users according to the numbers of enrolment by State[6] as shown in Table 4.3. We only determine the locations of users because *Twitter* users are not required to specify the age and gender in their profile.

TABLE 4.3: Minimum sample size for prediction model.

State	Enrolment	Twitter users in our dataset	Minimum sample size
New South Wales	4,816,991	13,471	349
Victoria	3,715,925	12,233	270
Queensland	2,840,091	5,360	206
Western Australia	1,452,272	2,630	105
South Australia	1,130,388	2,234	82
Tasmania	362,892	314	26
Australian Capital Territory	265,269	1,683	19
Northern Territory	128,971	268	10
Total	**14,712,799**	**38,193**	**1,067**

4.3.2 User's Vote Prediction

According to the voters' attitudes and opinion may be different depending on the specific political topic and the voters participating in different discussion events may have different political preference, our predicting model were computed based on the significance of sub-event topics and sentiment scores.

[6] http://results.aec.gov.au/17496/Website/GeneralEnrolmentByState-17496.htm

Definition 4.4. *A "voter preference" is defined as the highest sentiment score out of the two candidates. Given a user u and a set of messages M related to the two candidates with whom u has interactions, the aspect sentiment score is computed for each candidate.*

The sub-event score is calculated to evaluate the significance of each sub-event topic. The sub-event topic will have a high score if there is a lot of a message of them and many users discussing about it. In this work, sub-event score (SE_e) for a given event topic (e) is defined as:

$$SE_e = \frac{NoOfMessages_e}{NoOfTotalMessages} \times \frac{NoOfUsers_e}{NoOfTotalUsers} \tag{4.3}$$

All sub-events are ranked based on sub-event scores. In order to determine the voter preference among the candidates, for a given user we compute sentiment score for each candidate (i.e., "Abbott" and "Rudd"). For a given user, Aspect Sentiment (AS) scores are defined as Eq. 4.4 and 4.5 for "Tony Abbott" and "Kevin Rudd" respectively.

$$AS_{Abbott} = \frac{\sum_{m=1}^{pos}(asp\_score(m, Abbott) \times SE_m)}{\sum_{m=1}^{neg}(|asp\_score(m, Abbott)| \times SE_m)} \times \frac{C_{Abbott}}{C_{Abbott} + C_{Rudd}} \tag{4.4}$$

$$AS_{Rudd} = \frac{\sum_{m=1}^{pos}(asp\_score(m, Rudd) \times SE_m)}{\sum_{m=1}^{neg}(|asp\_score(m, Rudd)| \times SE_m)} \times \frac{C_{Rudd}}{C_{Abbott} + C_{Rudd}} \tag{4.5}$$

where $asp\_score(m, A)$ is the aspect-level score of message m, pos is the number of positive messages, neg is the number of negative messages, C_x is the number of both positive and negative messages of aspect x. If a given user posts only positive messages, we assign the summation of negative messages equal to 1. On the other hand, we assign the summation of positive messages equals to 1 when a user posts only negative messages. The voter preference is defined as the highest score out of the two candidates. If the scores are equal, we randomly selected the user vote. In addition, there is another possibility that people has negative sentiment while he still favour to the candidate however it is very difficult to identify.

$$UserVote_u = \begin{cases} \text{"}Abbott\text{"} & \text{if } AS_{Abbott} > AS_{Rudd} \\ \text{"}Rudd\text{"} & \text{if } AS_{Abbott} < AS_{Rudd} \\ Random(\text{"}Abbott\text{"}, \text{"}Rudd\text{"}) & \text{otherwise} \end{cases} \tag{4.6}$$

4.4 Experiments and Results

In this section, we firstly assess sub-event detection and sentiment analysis methods because both components may affect the final prediction results of our approach. Next, we evaluate our prediction results by computing the Mean Absolute Error (MAE) between the actual and predicted outcomes.

4.4.1 Dataset and Experimental Setting

A collection of messages posted by Australia-based users (given latitude, longitude and radius) via the *Twitter Search API* service from 4 August 2013 to 8 September 2013 with 808,661 messages with the user's initial event query is used for our experiments. We define an event by specifying the keyword query (i.e., *"#ausvotes13", "#election2013", "#AusVotes", "#auspol", "Kevin Rudd" and "Tony Abbott"*). We decided to choose this period because the election date is announced on 4 August 2013 and people started to discuss about this event. Also, we decided to choose the keywords related to the two candidates because as in Australian politics the candidates will be from the two major parties. Therefore, in this work we will predict the two-party-preferred vote.

For sub-event detection evaluation, we download the ground truth from *The Sydney Morning Herald* website in *Federal Politics* section[7]. It contains 115 real-world events during 4 August 2013 to 8 September 2013.

For sentiment analysis evaluation, we manually labelled 5,735 messages related to the first debate event of the 2013 Australian federal election, between *Kevin Rudd* and *Tony Abbott* on 11 August 2013 from 6pm to 9pm. There are 1,481 messages related to *Kevin Rudd* and 3,254 messages referring to *Tony Abbott*. The messages are annotated with a polarity score (positive, negative or neutral) and sarcasm by three local persons who have political knowledge. We assigned the message polarity score which was determined by the majority view of the three annotators.

For prediction evaluation, the messages since announce election day (i.e., 4 August 2013) until the day before election day (i.e., 6 September 2013) were used for predicting the results. We download the election results from *Australian Electoral Commission* website[8]. The two-party-preferred results for all states and territories as a national summary are compared. The four different national opinion

[7] http://www.smh.com.au/federal-politics/the-pulse-live
[8] http://www.aec.gov.au/Elections/Federal_Elections/2013/

polls are also compared with our results.

4.4.2 Baseline Approaches

In order to evaluate our approach for detecting sub-events in a collection of *tweets*, we compare our approach performance with temporal peaks detection approach in [66]. The authors bin the messages into a histogram by time (i.e., one hour in this work). Then, the authors calculate a historically weighted running average of message rate and identify rates that are significantly higher than the mean message rate. A window surrounding the local maximum is identified. Finally, top five frequent terms are presented as event name of each peak.

To evaluate our sentiment analysis method, we compare the performance of our method with aspect-based opinion summarization on *Twitter* data in the domain of politics. This work is introduced by Ringsquandl et al. in [88] which is the most similar work to ours. Researchers used the opinion lexicon which is presented in [116]. Semantic orientation of a word is the most probable class (positive, negative, neutral) of each opinion word according to *synsets* in *WordNet*. The final aspect-level sentiment is determined by a simple aggregation function which sums the semantic orientation of all words in the message that mentions the specific aspect.

Finally, we evaluate our prediction by comparing the performance of our approach with counting-based approaches [105] for our first baseline. For a second baseline, we adopt the idea from [93] by counting the number of *tweets* one week before the election day and using only the first message of each user for the prediction. However, we do not incorporate polls data in the second baseline. The third baseline is based on sentiment analysis only. We use the same size of our sample and the same algorithm of our sentiment analysis. We use the sum of sentiment scores for each aspect to predict the user votes. The third baseline is compared in order to see how well the combination between sub-event detection and sentiment analysis improve our results.

4.4.3 Evaluation

In this section, we evaluate the performance of our sub-event detection, sentiment analysis and the prediction approaches. For sub-event detection, we compare the precision, recall and F1-score against

the peak detection baseline.

$$Precision_{event} = \frac{\#detect\_realworld\_events}{\#total\_detect\_events}, \tag{4.7}$$

$$Recall_{event} = \frac{\#distinct\_detect\_realworld\_events}{\#total\_realworld\_events} \tag{4.8}$$

There is more than one detected event can relate to the same real-world event, then they are considered correct in terms of precision but only one event is considered in counting recall. In order to evaluate the performance of our sentiment analysis method, we compare the the $Precision$, $Recall$ and *F1-Score* of each polarity category against the aspect-based baseline.

$$Precision_{opinion} = \frac{T}{C}, \quad Recall_{opinion} = \frac{T}{L} \tag{4.9}$$

where T is the number of correct classified messages in one opinion category, C is the number of messages classified in one opinion category and L represents the number of the true labelled messages in one opinion category. Finally, we evaluate our prediction results by computing the Mean Absolute Error (MAE) between the actual and predicted outcomes.

Table 4.4 shows the $Precision$, $Recall$ and *F1-Score* of the sub-event detection of our approach against the peak detection baseline. In Table 4.4, we can observe that our approach can effectively detect real-world events which is significantly larger than the baseline. The baseline can detect smaller number of events because it considers only the temporal peaks in *tweet* frequency. Some events might not be frequently posted on social networks. On the other hand, our approach detects many duplicated events such as the first debate event. There are many different topics discussed during the debate which can cause many clusters when we perform the clustering process. However, our approach outperforms the baseline method by 31.43%. Table 4.5 represents the performance of the sentiment analysis of our approach against the baseline. It can be seen that our approach can effectively classify the micro-blog messages with a *F1-Score* which is significantly higher than the baseline in the same domain of politics.

Table 4.6 illustrates the performance of our prediction method against the three baselines. It can be seen that by incorporating sub-event detection and sentiment analysis can effectively improve the

TABLE 4.4: The performance of sub-event detection.

Method	# of detected events	# of real-life events	# of distinct real-life events	Precision (%)	Recall (%)	F1-Score (%)
Peak detection	19	14	14	73.68	12.17	20.89
Our approach	542	229	79	42.25	68.70	**52.32**

TABLE 4.5: The performance of sentiment analysis.

Aspect	Polarity	No. of Messages	Baseline (%)			Our approach (%)		
			Prec.	Rec.	F1	Prec.	Rec.	F1
Kevin Rudd (ALP)	Positive	327	32.72	37.15	34.80	70.34	47.92	**57.00**
	Negative	726	18.87	70.98	29.82	54.41	83.51	**65.89**
	Neutral	428	79.44	34.00	47.62	67.76	54.92	**60.67**
Tony Abbott (LNP/Coalition)	Positive	334	38.92	22.03	28.14	62.28	31.09	**41.48**
	Negative	1,624	22.84	72.89	34.79	59.05	74.75	**65.98**
	Neutral	1,296	76.47	45.99	57.43	62.89	62.60	**62.74**

prediction accuracy in both state and national levels. In addition, it can correctly predict five out of eight states and territories with smallest error and only 4.43% error for national level. Table 4.7 presents the performance of our approach against the four different national opinion polls. It can observe that our method comes close to traditional polls with the same trend.

TABLE 4.6: MAE for comparing election results with three baselines (%).

State	Election result		Baseline1		Baseline2		Baseline3		Our method	
	ALP	LNP	ALP	MAE	ALP	MAE	ALP	MAE	ALP	MAE
NSW	45.65	54.35	37.94	7.71	44.81	0.84	42.60	3.05	43.11	2.54
VIC	50.20	49.80	37.11	13.09	40.41	9.79	39.00	11.20	38.48	11.72
QLD	43.02	56.98	42.44	0.58	51.35	8.33	45.05	2.03	45.56	2.54
WA	41.72	58.28	37.11	4.61	44.80	3.08	41.38	0.34	41.02	0.70
SA	47.64	52.36	33.21	14.43	46.88	0.76	40.94	6.70	42.38	5.26
TAS	51.23	48.77	26.35	24.88	35.00	16.23	35.11	16.12	38.40	12.83
ACT	59.91	40.09	38.23	21.68	46.58	13.33	42.61	17.30	45.54	14.37
NT	49.65	50.35	35.11	14.54	58.06	8.41	38.08	11.57	42.74	6.91
Average				12.69		7.60		8.54		**7.11**
National	46.51	53.49	37.23	9.28	55.64	9.13	41.69	4.82	42.08	**4.43**

4.5 Summary

In our study, the incorporating between sub-event detection and sentiment analysis achieved better prediction results than the three baselines. It might suggest that the discussions of sub-event topics

TABLE 4.7: MAE for comparing election results (National) with opinion polls (%).

Firm	Date	ALP	LNP	MAE	Remark
Morgan (multi) [89]	4-6 Sep 2013	46.50	53.50	1.01	
ReachTEL [86]	5 Sep 2013	47.00	53.00	0.49	
Newspoll [75]	3-5 Sep 2013	46.00	54.00	0.51	excludes Northern Territory
Essential [22]	1-4 Sep 2013	48.00	52.00	1.49	
Our approach		42.08	57.92	4.43	

that user had engaged in influenced their voting. Also, it can be seen that *Twitter* is able to reflect underlying trend in a political campaign. Even if people who use social media are not completely representative of the public, the amount of attention paid to an issue is an indicator of what is happening in society. Our approach allows researchers to surface user opinions of the social sphere at different time points to determine a view of sentiment for a given event. Also, it turns out that what people say on *Twitter* is a very good indicator of how they will vote.

In this work, we studied a problem of predicting elections based on publicly available data on social networks, like *Twitter*. An effective method of predicting election results is proposed. An approach to detecting sub-events and performing sentiment analysis over micro-blogs in order to predict user preferences is also presented. Extensive experiments are conducted to have evaluated the performance of our approach on a real-world *Twitter* dataset. The proposed approach is effective in predicting election results against the given baselines and comes close to the results of traditional polls. In future work, we will further consider the sarcasm identification and analysis. More studies on the credibility will be conducted in order to remove disinformation and spamming.

Chapter 5

Invariant Event Tracking in Social Networks

5.1 Problems and Challenges

When an event is emerging and actively discussed on social networks, its related issues may change from time to time. People may focus on different issues of an event at different times. We define an invariant event as follows:

Definition 5.1. *An "invariant event" is an event with changing subsequent issues that last for a period of time.*

Examples of invariant events include government elections, natural disasters, and breaking news. The monitoring of events over social networks has many applications such as decision making and situation awareness. As a particular event develops, people may be interested in seeing an overview of the situation. An event may have several related topics that develop over time.

In this chapter, we introduce a new concept called invariant event tracking. An event is a social activity or a phenomena that occurs in a certain place during a certain time period. Event tracking is to monitor streams of topic-discussions in order to understand the event. A series of changing topics derived from an event over time is called an invariant event. In general, a topic is associated with a set of keywords. At any point in time, there are multiple topics discussed on social networks. Invariant event tracking is important for analyzing the overall situation of a particular event on social networks. For example, during a natural disaster, government may need to analyze the development of situations in order to make the right responses at the right times. For a longer-running event like a government

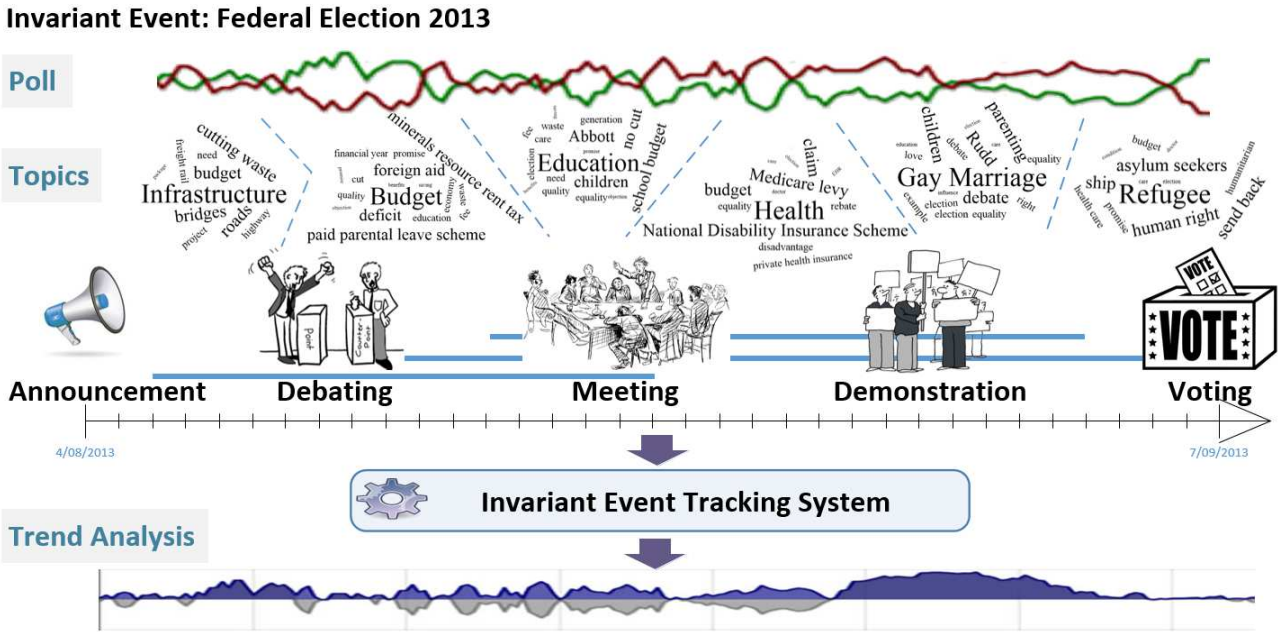


FIGURE 5.1: Invariant event tracking conceptual diagram.

election, people may wish to track the event with-respect-to multiple issues such as campaign-launch speeches and a number of open TV-debates under different topics, in order to cast their votes.

However, general micro-blog searches for given keywords return large amounts of messages that are not grouped or organized in any meaningful way. It is difficult for people to comprehend a large number of messages in a chronological order and to monitor an event as it unfolds. Although traditional techniques such as clustering are able to capture major events in social networks [114, 56, 108], it is difficult to capture the incidental events that may or may not be relevant to the current focused event. For example, when a natural disaster has just occurred, people may initially talk about the natural phenomenon that they have just witnessed. Then, damages, casualties, or the consequences of the disaster might be reported. Topics related to volunteer organizations and rescue activities might also be discussed later. All these topics are related to the same event, yet a general clustering approach is not able to correlate them into a single event.

In this work, we propose an approach of invariant event tracking on social networks. We use our system to track an event based on micro-blog messages and monitor the topic changes over time for an event that is rendered to the system as a set of keywords. The research challenges are: (1) effectively summarizing the given event-search query (termed as an invariant event), and (2) tracking the evolution of an event within a given time period.

5.2 Invariant Event Tracking Approach

In order to show a comprehensive understanding of our invariant event tracking framework, a conceptual diagram is presented in Figure 5.1. The architecture of our system consists of three components, including micro-blog loader and pre-processing, invariant event tracking and event visualization as shown in Figure 5.2. The following information provides details of each component.

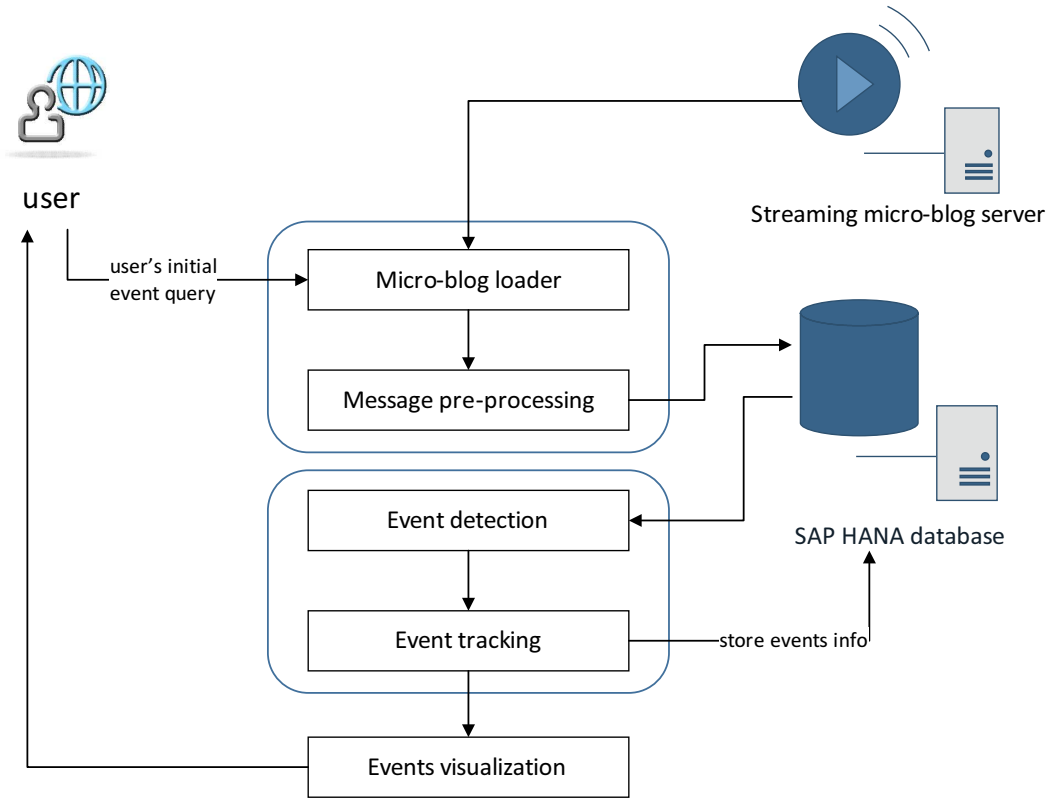


FIGURE 5.2: The architecture of our system.

5.2.1 Micro-blog Loader and Pre-processing

A micro-blog loader is developed to collect the *Twitter* messages from public users via the Java library API service. The user's initial query (i.e., a set of keywords) is used for specifying an event. To support high arrival rates of incoming micro-blogs, we take advantage of In-Memory database technology namely High-Performance Analytic Appliance (*SAP HANA* Database[1]) to process both transactional and analytical workloads fully in-memory. The pre-processing was designed to ignore

[1] http://www.saphana.com

common words that carry less important meaning than keywords and to remove irrelevant data such as *re-tweet* keyword and web address. We utilized *SAP HANA* text analysis processing by using the two predefined configurations; *LINGANALYSIS_FULL* (Linguistic Analysis) and *EXTRACTION_CORE*. The *LINGANALYSIS_FULL* is used for segmentation, stemming and tagging words' parts of speech. The *EXTRACTION_CORE* is the process of discovering and presenting specific entities such as names of people, places, things, dates and values. The stop words are also removed. Example of *SAP HANA* predefined configuration for segmentation and stemming is show as follows:

```
CREATE FullText INDEX
  "TWEETS_FTI_LINGANALYSIS_FULL"
  ON SAYAN."ELECTION2013_AUS"("StatusText1")
  TEXT ANALYSIS ON
  CONFIGURATION 'LINGANALYSIS_FULL';
```

Example of *SAP HANA* predefined configuration for entities extraction is show as follows:

```
CREATE FullText INDEX
  "TWEETS_FTI_CORE"
  ON SAYAN."ELECTION2013_AUS"("StatusText2")
  TEXT ANALYSIS ON
  CONFIGURATION 'EXTRACTION_CORE';
```

5.2.2 Invariant Event Tracking

In this task, we tackle the issues of event detection and tracking in social networks. Event detection is to identify hierarchically nested event topics that break down an event into more refined parts. Then, event tracking will be performed to discover an invariant event.

Event detection: in order to determine hierarchically nested events, we aim to group the co-occurring keywords for topic discovery. Note that the concepts of event and topic are different; an

event may have several topics at different stages in its life cycle. We adopt the idea of community detection in graphs for locating and analyzing overlapping dense user groups in social networks [79, 102], where the communities can be formed in terms of user common interests. In our approach, a so-called community that represents a set of users is termed as an episode that includes the topics related to an event at a certain time frame (see Figure 5.3). Therefore, in our approach, the migration of members amongst communities is treated as the evolution of the topics amongst different episodes in an invariant event. The computation is based on the Clique Percolation Method [79]. We borrow this idea because the keywords of topics can appear in more than one event. We partition the messages into time frames. The size of time frame is defined by time interval according to user preference (e.g., one day in our experiment). For each time frame, co-occurring keywords that appear together in at least min_occur are extracted. To compute co-occurring keywords, we exclude *re-tweet* messages. Networks of keywords are then constructed as graphs. Finally, the keywords in an episode are grouped along with the topics of the episode. Each episode represents one or more event topics in a particular time frame.

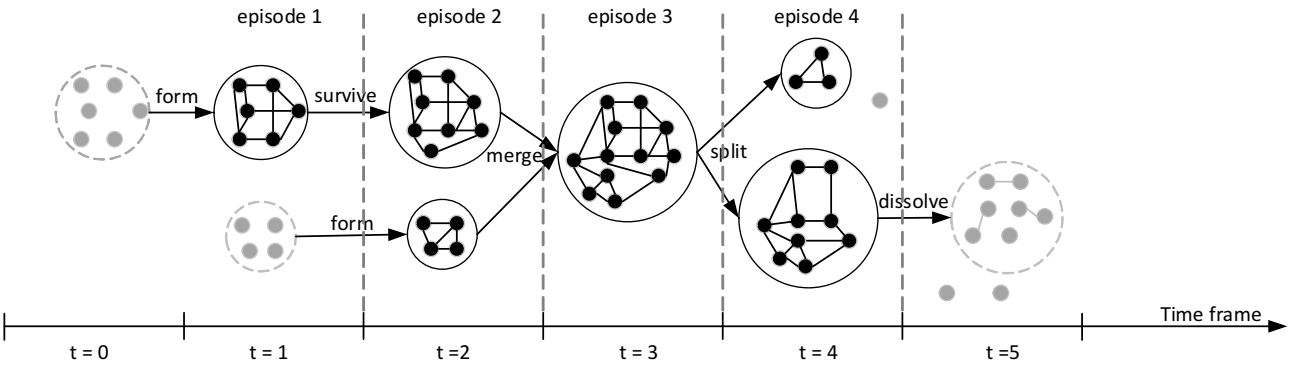


FIGURE 5.3: Example of topic changes over time frame.

Event tracking: at this stage, we aim to identify an invariant event by tracking all the event topics detected at each time frame. The event evolution is detected at different time frames. We model an invariant event as a graph sequence as follows:

Definition 5.2. *An "invariant event" is the set of event topics ($T_i = \{T_i^1, T_i^2, ..., T_i^n\}$) denotes the n topics detected at the i^{th} time frame, where topic T_i^k is also a graph represented by (V_i^k, E_i^k). The*

node set is denoted by V to represent keywords in the messages and edge set is denoted by E to represent the co-occurrence between keywords at the i^{th} time frame.

The event evolution is represented by a series of episodes from different time frames. In order to capture the changes of episodes, we consider five types of transitions (i.e., form, dissolve, survive, split, and merge) [102]. At time frame t, we construct a weighted bipartite graph between topics at $t-1$. The weight between sets of topic keywords is computed by the similarity between the groups of keywords belonging to different topics. Two groups of topic keywords are matched if at least min_match percent of their keywords are the same. However, the topic evolution in social networks is different from the evolution of social communities. On average 72 percent of topics each day are new in *Twitter* [50]. Thus, we only compare between the groups of keywords obtained at current time frame t and the groups of keywords at previous time frame $t-1$. The similarity of topics that have different sets of keywords is defined as follows:

$$topic\_sim(T_t^a, T_{t-1}^b) = \frac{|V_t^a \cap V_{t-1}^b|}{max(|V_t^a|, |V_{t-1}^b|)} \tag{5.1}$$

Topic evolution is a sequence of changes succeeding each other in the consecutive time frame. Based on the idea of detecting evolution of communities [102], we define the transitions of topics as follows:

- **Form**: A new topic forms when it did not exist in the previous time frame $(t-1)$ but it appears in the current time frame (t).

- **Dissolve**: A topic in the previous time frame $(t-1)$ dissolves when it does not occur in the current time frame (t) at all.

- **Survive**: A topic survives when two groups in the consecutive time frames are matched. It can be **continuing** (i.e., two topics differ only by few keywords but their size remains the same), **growing** (i.e., some new keywords have joined the group) and **shrinking** (i.e., some keywords have left the group).

  Continuing : $topic\_sim(T_t^a, T_{t-1}^b) > min\_match$ and $|T_{t-1}^b| = |T_t^a|$

Growing : $topic\_sim(T_t^a, T_{t-1}^b) > min\_match$ and $|T_{t-1}^b| < |T_t^a|$

Shrinking : $topic\_sim(T_t^a, T_{t-1}^b) > min\_match$ and $|T_{t-1}^b| > |T_t^a|$

- **Split**: A topic splits into two or more topics in the next time frame when some topics from current time frame (t) consist of keywords of one topic from the previous time frame $(t-1)$.

  $\frac{|(V_t^1 \cup V_t^2 ... \cup V_t^k) \cap V_{t-1}^b|}{|V_{t-1}^b|} > min\_match$ where

  $T_t^* = \{T_t^1, ..., T_t^k\} \in T_t^n$ and $k > 1$

  $\forall T_t^a \in T_t^*, \frac{|V_t^a \cap V_{t-1}^b|}{|V_t^a|} > min\_match$

- **Merge**: A set of topics merges into the same topic in the next time frame when some topics from previous time frame $(t-1)$ consist of keywords of one topic from the current time frame (t).

  $\frac{|V_t^a \cap (V_{t-1}^1 \cup V_{t-1}^2 ... \cup V_{t-1}^k)|}{|V_t^a|} > min\_match$ where

  $T_{t-1}^* = \{T_{t-1}^1, ..., T_{t-1}^k\} \in T_{t-1}^n$ and $k > 1$

  $\forall T_{t-1}^b \in T_{t-1}^*, \frac{|V_t^a \cap V_{t-1}^b|}{|V_{t-1}^b|} > min\_match$

All event topics, which are linked together over time frames, are represented as an invariant event. The example of topic changes over time frame is shown in Figure 5.3. For each time frame, node is the keyword and edge between the nodes is formed when those keywords co-occur in at least min_occur times.

5.2.3 Event Visualization

We designed a dashboard to display an invariant event and topic evolution over time. Events are presented via *Annotated Time Line Chart* [2] as show in Figure 5.4 (top-left) for each day (represented by letters A to Z). For a given invariant event, the size of *Timecloud* indicates the frequency of words over the selected time period as show in Figure 5.5. The color represents the sentiment analysis of the words [109], in which the more darker color the word is, the more positive attitude that the related messages are of. Figure 5.5(a) shows the most discussed words and the messages' attitudes

[2] https://google-developers.appspot.com/chart/interactive/docs/gallery/annotatedtimeline

towards them over the entire time. Figure 5.5(b) shows the most frequently used words and their related messages' attitudes for a particular day. It also shows the evolution of the conversation through the appearance and disappearance of those keywords in Figure 5.5(c). For *Timecloud*, we applied *Tagscloud*[3] which is a combination of word clouds with a chronological dimension. Original messages are shown in Figure 5.4 (bottom-left). *Stream graph*, which is a visualization for displaying multiple time series, is used to show the amount of people using the words over time as shown in Figure 5.4 (bottom-right).

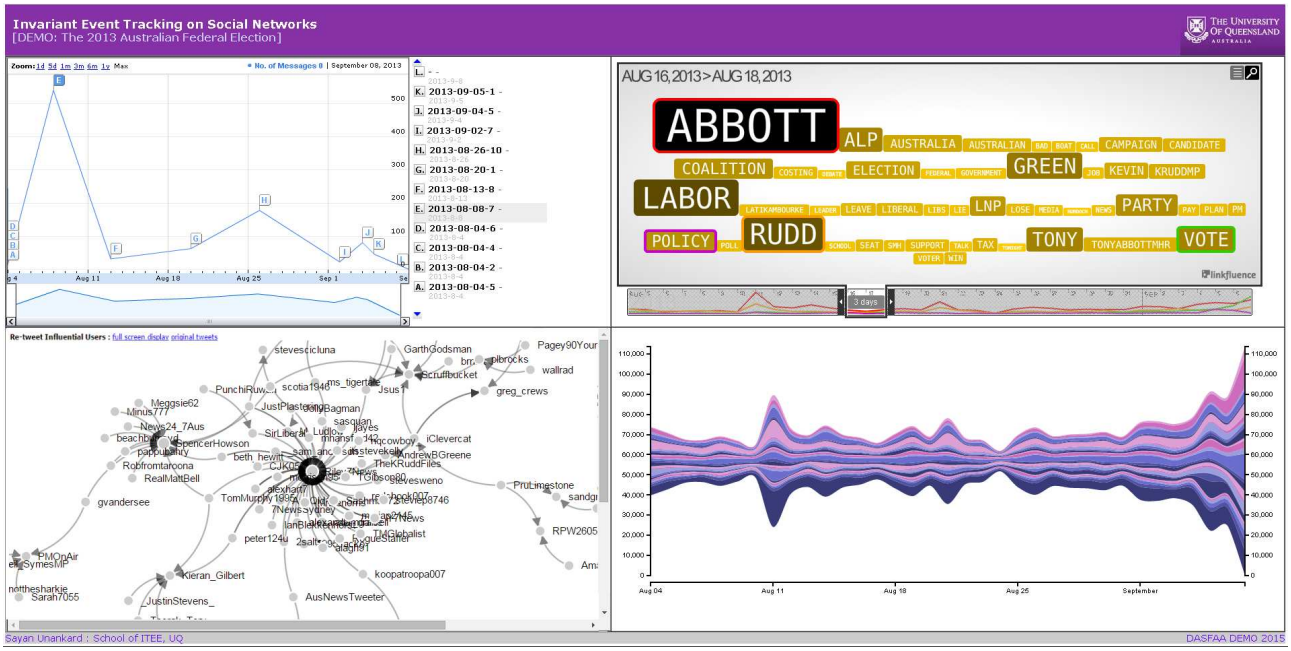


FIGURE 5.4: Dashboard of the system.

(a) for the selected time period (b) for the selected date (c) the evolution of the conversation

FIGURE 5.5: Screen displays of the TimeCloud.

[3] http://widget.rtgi.eu/public/tagclouds/

5.3 Demonstration Scenario

For demonstration, a collection of messages posted by Australian locals (given latitude, longitude and radius) via the *Twitter Search API* service from 4 August 2013 to 8 September 2013 with 808,661 messages with the user's initial event-query, is used. We define an event by specifying the keyword query (i.e., *"#ausvotes13", "#election2013", "#AusVotes", "#auspol", "Kevin Rudd" and "Tony Abbott"*). We decided to choose this period because the election date was announced on 4 August 2013 and people started discussions on this event. Also, we chose the keywords related to the two candidates because in Australian politics the candidates will be from the two major parties. We developed the system using Java technology and stored data on *SAP HANA* in-memory database.

For a user, the system displays all event topics for each day. A user can track an event by selecting the event topics in a particular day (represented by letters A to Z). The *Timecloud* of the selected event topics will be displayed. In the example of Figure 5.5, it shows a *Timecloud* for a given event topic (i.e., Mr Tony Abbott made the announcement regard homosexuality in Sydney to launch their policy). This event topic was discussed between the days of 29 August 2013 and 30 August 2013. Figure 5.5(a) provides the most discussed words all the time for this event topic. In addition, user can change the range of days to cover a different period of time.

Furthermore, it is also possible to toggle the LIST command-button (i.e., the icon in the upper right hand corner, next to the magnifying glass) to display all the words with the same size, including unused words (in light grey) for the selected period as shown in Figure 5.5(c). When selecting a word on the *Timecloud*, a graph will appear at the bottom, showing relative volume of mentioning over the entire timeline. In Figure 5.4 (bottom-right), the stream graph displays changes in the magnitude of a topic based on the activity level (number of words) over a given time period.

5.4 Summary

In this work, we proposed an approach to tracking invariant events and topic evolution within a given time period. The main contributions of this work are twofold. (1) An effective approach of tracking

invariant events is proposed by incorporating CPM community mining and community evolution discovery techniques. (2) We have implemented an invariant event tracking system which provides user with an overview of the development of an event. The system supports event tracking by allowing users to specify the time period in order to visualize the words consequently appearing and disappearing over time. We used the *Twitter* messages related to the 2013 Australian Federal Election to demonstrate the effectiveness of our approach. The further performance evaluation will be conducted in our future work.

Chapter 6

Conclusions and Future Works

6.1 Conclusions

This research is driven by the research questions, "how can we detect and track emerging events from social networks?". The main objectives of the research have been successfully achieved. We are able to provide data mining approaches for tackling event detection and tracking challenges in a real-world scenario. In this thesis, we have accomplished three tasks: (1) identified subsistent problems and challenges in event detection in streaming micro-blog text, (2) developed effective solutions to address these problems, and (3) evaluated the proposed approaches using real-world streaming datasets.

In Chapter 1, we first introduced the research background and motivation of emerging event detection in social networks, including the definitions of an event and its analysis in social networks. In particular, we presented a background of micro-blog data and events in social networks. Then we conducted statistical studies to understand how information spreads through large user communities. Finally, we presented research problems, challenges, and contributions to this thesis.

In Chapter 2, we presented a brief literature review on the research topics that are related to our work. We divided the literature review into five sections. Firstly, we reviewed research works on message propagation in social networks and presented our preliminary work to predict the *re-tweet* activity of any given message for a particular event. Secondly, we summarised recent research progresses on event detection methods. Thirdly, we presented short-text clustering methods, which were utilised for our event detection models. Then, the existing works related to event tracking

were introduced. Finally, we presented a brief summary of sentiment analysis on social networks for extracting users' opinions from short text messages.

In Chapter 3, we proposed an approach for the early detection of emerging hotspot events in social networks with location sensitivity. We considered the message-mentioned locations for identifying the locations of events. Our approach has five stages. Firstly, the pre-processing is performed to remove stop words and irrelevant data. An algorithm is designed for slang conversion, synonym expansion and conceptual similarity to provide a rich semantic context for measuring message similarity to improve clustering results. Secondly, we propose a clustering approach to automatically group the messages into events. Our approach requires no prior knowledge of the number of events. Therefore, hierarchical clustering is used in our approach. Thirdly, we propose a hotspot event detection method. Fourthly, emerging hotspot event detection is performed. We identify strong correlations between user locations and event locations in detecting the emerging events. Finally, we develop a visualization model for representing emerging events. We evaluate our approach based on a real-world *Twitter* dataset. Our experiments show that the proposed approach can effectively detect emerging events with respect to user's locations that have different granularities. Our approach can achieve high *Precision* score because it can filter out non-real world event clusters. Most of non-real world event clusters do not contain event locations. For example, the cluster which contains *hashtag "#FF" (i.e., FollowFriday)* is emerging only on Friday and it will be removed from our approach because it does not contain the event location mentioned in the cluster.

In Chapter 4, we proposed an approach for sub-event detection and sentiment analysis for a given long-running event. For a long-running event like a nation-wide election which usually has fixed start and end times, users may want to monitor sub-events such as the debate or campaign launch speech. Alternatively, policy-makers may want to know the feeling of users during the course of an election. In order to provide a complete coverage of sub-event detection and sentiment analysis in social networks, we proposed an approach which has four stages. Firstly, pre-processing is performed to remove irrelevant data from the dataset. Secondly, we conduct a clustering approach to automatically group the messages into sub-events. Thirdly, we propose a lexicon-based approach to detect users' opinions for specific entities. Finally, we develop a visualization model for representing sub-events and users' opinions.

To evaluate our approach we presented an approach to detect users' political preferences and predict the election results by incorporating sub-event detection and sentiment analysis at a state as well as a national level, as a case study. For sub-event detection approach, we can observe that our approach can effectively detect real-world events which is significantly larger than the baseline by 31.43%. The baseline can detect smaller number of events because it considers only the temporal peaks in *tweet* frequency. Some events might not be frequently posted on social networks. For sentiment analysis, it can be seen that our approach can effectively classify the micro-blog messages with a *F1-Score* which is significantly higher than the baseline in the same domain of politics. For prediction approach, our approach achieved better prediction results than the given baselines and comes close to the results of traditional polls. It might suggest that the discussions of sub-event topics that users had engaged in had influenced their voting. Also, it can be seen that *Twitter* is able to reflect underlying trends in a political campaign.

In Chapter 5, with the variety of events discussed in micro-blog, people may be interested in understanding the whole situation of an event. We introduced an invariant event tracking system, which is focused on analysing the continuous invariant events and their movements in a particular time period. We detect events by utilizing the Clique Percolation Method (CPM) community mining and track invariant event based on the relationships between communities. To demonstrate our approach, we use the *Twitter* messages related to the 2013 Australian federal election event with a given set of keywords search retrieved from the announced election day until the day after election day. The results show that our approach can capture the development of event for a given time period.

6.2 Future Directions

In this section, we propose the following two possible directions for future work that have potential for further investigation. It is hoped that the research presented in this thesis will lead to a deeper understanding of event detection and tracking in social networks, and will encourage constructive future work in this area.

6.2.1 A Storyboard for Event Summarization

For a long-running event discussed in micro-blog, people may be interested in understanding the whole situation of an event. Users may need a summary of all occurrences to date. We will explore the problem of generating storyboards from micro-blogs for user's initial input queries. This problem is challenging because of the sparse, dynamic and social nature of micro-blogs for providing both superior user experience and deeper understanding of real-time events. It would be helpful for anyone in monitoring the event's evolution.

6.2.2 Finding Story Chains in Social Networks

The large amount of unstructured search results returned by social network search engines makes it hard to track the evolution of an event. For complex events, users may not be able to see the big picture of the story (i.e., a sequence of events). The information that the users get from social network search engines would be more informative if we could uncover the hidden relationships between events.

References

[1] F. Abel, C. Hauff, G.-J. Houben, R. Stronkman, and K. Tao. Semantics + filtering + search = twitcident. exploring information in social web streams. In *23rd ACM Conference on Hypertext and Social Media, HT '12, Milwaukee, WI, USA, June 25-28, 2012*, pages 285–294, 2012.

[2] C. C. Aggarwal and K. Subbian. Event detection in social streams. In *Proceedings of the Twelfth SIAM International Conference on Data Mining, Anaheim, California, USA, April 26-28, 2012.*, pages 624–635, 2012.

[3] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia*, pages 37–45, 1998.

[4] F. Alvanaki, S. Michel, K. Ramamritham, and G. Weikum. See what's enblogue: real-time emergent topic identification in social media. In *15th International Conference on Extending Database Technology, EDBT '12, Berlin, Germany, March 27-30, 2012, Proceedings*, pages 336–347, 2012.

[5] S. Banerjee and T. Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *Computational Linguistics and Intelligent Text Processing, Third International Conference, CICLing 2002, Mexico City, Mexico, February 17-23, 2002, Proceedings*, pages 136–145, 2002.

[6] S. Banerjee, K. Ramanathan, and A. Gupta. Clustering short texts using wikipedia. In *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and*

Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007, pages 787–788, 2007.

[7] H. Becker, M. Naaman, and L. Gravano. Learning similarity metrics for event identification in social media. In *Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010, New York, NY, USA, February 4-6, 2010*, pages 291–300, 2010.

[8] H. Becker, M. Naaman, and L. Gravano. Beyond trending topics: Real-world event identification on twitter. In *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*, pages 438–441, 2011.

[9] F. Benevenuto, T. Rodrigues, M. Cha, and V. A. F. Almeida. Characterizing user behavior in online social networks. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement 2009, Chicago, Illinois, USA, November 4-6, 2009*, pages 49–62, 2009.

[10] D. Boyd, S. Golder, and G. Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *43rd Hawaii International International Conference on Systems Science (HICSS-43 2010), Proceedings, 5-8 January 2010, Koloa, Kauai, HI, USA*, pages 1–10, 2010.

[11] C. Budak, D. Agrawal, and A. El Abbadi. Structural trend analysis for online social networks. *PVLDB*, 4(10):646–656, 2011.

[12] M. Cataldi, L. D. Caro, and C. Schifanella. Emerging topic detection on twitter based on temporal and social terms evaluation. In *Proceedings of the Tenth International Workshop on Multimedia Data Mining*, pages 4:1–4:10, 2010.

[13] M. Cha, A. Mislove, and P. K. Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pages 721–730, 2009.

[14] L. Chen and A. Roy. Event detection from flickr data through wavelet-based spatial analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009*, pages 523–532, 2009.

[15] L. Chen, W. Wang, M. Nagarajan, S. Wang, and A. P. Sheth. Extracting diverse sentiment expressions with target-dependent polarity from twitter. In *Proceedings of the Sixth International Conference on Weblogs and Social Media, Dublin, Ireland, June 4-7, 2012*, pages 50–57, 2012.

[16] Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 759–768, 2010.

[17] W. G. Cochran. *Sampling techniques*. Wiley, New York, 1977.

[18] A. Cui, M. Zhang, Y. Liu, S. Ma, and K. Zhang. Discover breaking events with popular hashtags in twitter. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 1794–1798, 2012.

[19] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.

[20] X. Ding, B. Liu, and P. S. Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the International Conference on Web Search and Web Data Mining, WSDM 2008, Palo Alto, California, USA, February 11-12, 2008*, pages 231–240, 2008.

[21] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification and Scene Analysis*. John Wiley & Sons Inc, 2nd edition, 2001.

[22] Essential. Two party preferred, federal politics voting intention. `http://essentialvision.com.au/documents/essential_report_130905.pdf`. Accessed: 2014-07-07.

[23] H. Fang. A re-examination of query expansion using lexical resources. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 139–147, 2008.

[24] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu. Parameter free bursty events detection in text streams. In *Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, August 30 - September 2, 2005*, pages 181–192, 2005.

[25] W. Galuba, K. Aberer, D. Chakraborty, Z. Despotovic, and W. Kellerer. Outtweeting the twitterers - predicting information cascades in microblogs. In *3rd Workshop on Online Social Networks, WOSN 2010, Boston, MA, USA, June 22, 2010*, pages 1–9, 2010.

[26] D. Gayo-Avello. I wanted to predict elections with twitter and all i got was this lousy paper - a balanced survey on election prediction using twitter data. *Computing Research Repository*, abs/1204.6441:1–13, 2012.

[27] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - Short Papers*, pages 42–47, 2011.

[28] R. González-Ibáñez, S. Muresan, and N. Wacholder. Identifying sarcasm in twitter: A closer look. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - Short Papers*, pages 581–586, 2011.

[29] S. Goorha and L. H. Ungar. Discovery of significant emerging trends. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010*, pages 57–64, 2010.

[30] D. Gruhl, R. V. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004*, pages 491–501, 2004.

[31] A. Guille and C. Favre. Mention-anomaly-based event detection and tracking in twitter. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2014, Beijing, China, August 17-20, 2014*, pages 375–382, 2014.

[32] M. Gupta, J. Gao, C. Zhai, and J. Han. Predicting future popularity trend of events in microblogging platforms. *ASIST*, 49(1):1–10, 2012.

[33] L. Hong, O. Dan, and B. D. Davison. Predicting popular messages in twitter. In *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011 (Companion Volume)*, pages 57–58, 2011.

[34] A. Hotho, S. Staab, and G. Stumme. Ontologies improve text document clustering. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), 19-22 December 2003, Melbourne, Florida, USA*, pages 541–544, 2003.

[35] J. Hu, L. Fang, Y. Cao, H.-J. Zeng, H. Li, Q. Yang, and Z. Chen. Enhancing text clustering by leveraging wikipedia semantics. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008*, pages 179–186, 2008.

[36] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, pages 168–177, 2004.

[37] X. Hu, J. Tang, H. Gao, and H. Liu. Unsupervised sentiment analysis with emotional signals. In *WWW*, pages 607–618, 2013.

[38] A.-L. Huang, D. N. Milne, E. Frank, and I. H. Witten. Clustering documents using a wikipedia-based concept representation. In *Advances in Knowledge Discovery and Data Mining, 13th Pacific-Asia Conference, PAKDD 2009, Bangkok, Thailand, April 27-30, 2009, Proceedings*, pages 628–636, 2009.

[39] G. Huang, J. He, Y. Zhang, W. Zhou, H. Liu, P. Zhang, Z. Ding, Y. You, and J. Cao. Mining streams of short text for analysis of world-wide event evolutions. *World Wide Web*, pages 1–17, 2014.

[40] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Micro-blogging as online word of mouth branding. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Extended Abstracts Volume, Boston, MA, USA, April 4-9, 2009*, pages 3859–3864, 2009.

[41] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Twitter power: Tweets as electronic word of mouth. *JASIST*, 60(11):2169–2188, 2009.

[42] A. Java, X. Song, T. Finin, and B. L. Tseng. Why we twitter: An analysis of a microblogging community. In *Advances in Web Mining and Web Usage Analysis, 9th International Workshop on Knowledge Discovery on the Web, WebKDD 2007, and 1st International Workshop on Social Networks Analysis, SNA-KDD 2007, San Jose, CA, USA, August 12-15, 2007. Revised Papers*, pages 118–138, 2007.

[43] A. Joshi, B. A. R., P. Bhattacharyya, and R. K. Mohanty. C-feel-it: A sentiment analyzer for micro-blogs. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - System Demonstrations*, pages 127–132, 2011.

[44] A. Jungherr, P. Jurgens, and H. Schoen. Why the pirate party won the german election of 2009 or the trouble with predictions: A response to tumasjan, a., sprenger, t. o., sander, p. g., & welpe, i. m. "predicting elections with twitter: What 140 characters reveal about political sentiment". *Social Science Computer Review*, 30(2):229–234, May 2012.

[45] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, volume 163. Prentice Hall, 2000.

[46] K. Kim, K. Zettsu, Y. Kidawara, and Y. Kiyoki. Moving phenomenon: Aggregation and analysis of geotime-tagged contents on the web. In *Web and Wireless Geographical Information Systems, 9th International Symposium, W2GIS 2009, Maynooth, Ireland, December 7-8, 2009. Proceedings*, pages 7–24, 2009.

[47] K.-S. Kim, R. Lee, and K. Zettsu. mtrend: Discovery of topic movements on geo-microblogging messages. In *19th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2011, November 1-4, 2011, Chicago, IL, USA, Proceedings*, pages 529–532, 2011.

[48] B. Krishnamurthy, P. Gill, and M. F. Arlitt. A few chirps about twitter. In *Proceedings of the first Workshop on Online Social Networks, WOSN 2008, Seattle, WA, USA, August 17-22, 2008*, pages 19–24, 2008.

[49] G. Kumaran and J. Allan. Text classification and named entities for new event detection. In *SIGIR 2004: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, July 25-29, 2004*, pages 297–304, 2004.

[50] H. Kwak, C. Lee, H. Park, and S. B. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 591–600, 2010.

[51] C. Lee, H. Yang, T. Chien, and W. Wen. A novel approach for event detection by mining spatio-temporal information on microblogs. In *International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2011, Kaohsiung, Taiwan, 25-27 July 2011*, pages 254–259, 2011.

[52] P. Lee, L. V. S. Lakshmanan, and E. E. Milios. Keysee: supporting keyword search on evolving events in social streams. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, pages 1478–1481, 2013.

[53] K. Lerman and R. Ghosh. Information contagion: An empirical study of the spread of news on digg and twitter social networks. In *Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA, May 23-26, 2010*, pages 90–97, 2010.

[54] J. Leskovec, L. Backstrom, and J. M. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 497–506, 2009.

[55] J. Letierce, A. Passant, S. Decker, and J. G. Breslin. Understanding how twitter is used to spread scientific messages. In *Proceedings of the WebSci10: Extending the Frontiers of Society On-Line, April 26-27th, 2010, Raleigh, NC: US*, pages 1–8, 2010.

[56] C. Li, A. Sun, and A. Datta. Twevent: segment-based event detection from tweets. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 155–164, 2012.

[57] R. Li, K. H. Lei, R. Khadiwala, and K. C. Chang. TEDAS: A twitter-based event detection and analysis system. In *IEEE 28th International Conference on Data Engineering (ICDE 2012), Washington, DC, USA (Arlington, Virginia), 1-5 April, 2012*, pages 1273–1276, 2012.

[58] X. Li, H. Cai, Z. Huang, Y. Yang, and X. Zhou. Spatio-temporal event modeling and ranking. In *Web Information Systems Engineering - WISE 2013 - 14th International Conference, Nanjing, China, October 13-15, 2013, Proceedings, Part II*, pages 361–374, 2013.

[59] Z. Li, B. Wang, M. Li, and W. Ma. A probabilistic model for retrospective news event detection. In *SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, August 15-19, 2005*, pages 106–113, 2005.

[60] C. Lin, C. Lin, J. Li, D. Wang, Y. Chen, and T. Li. Generating event storylines from microblogs. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 175–184, 2012.

[61] B. Liu and L. Zhang. A survey of opinion mining and sentiment analysis. In C. C. Aggarwal and C. Zhai, editors, *Mining Text Data*, pages 415–463. Springer US, 2012.

[62] S. Liu, F. Liu, C. T. Yu, and W. Meng. An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In *SIGIR 2004: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, July 25-29, 2004*, pages 266–272, 2004.

[63] B. D. Longueville, R. S. Smith, and G. Luraschi. "omg, from here, I can see the flames!": a use case of mining location based social networks to acquire spatio-temporal data on forest fires. In *Proceedings of the 2009 International Workshop on Location Based Social Networks, LBSN 2009, November 3, 2009, Seattle, Washington, USA, Proceedings*, pages 73–80, 2009.

[64] A. Makazhanov and D. Rafiei. Predicting political preference of twitter users. In *Advances in Social Networks Analysis and Mining 2013, ASONAM '13, Niagara, ON, Canada - August 25 - 29, 2013*, pages 298–305, 2013.

[65] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller. Tweets as data: demonstration of tweeql and twitinfo. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2011, Athens, Greece, June 12-16, 2011*, pages 1259–1262, 2011.

[66] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller. Twitinfo: aggregating and visualizing microblogs for event exploration. In *Proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011, Vancouver, BC, Canada, May 7-12, 2011*, pages 227–236, 2011.

[67] M. Mathioudakis and N. Koudas. Twittermonitor: trend detection over the twitter stream. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2010, Indianapolis, Indiana, USA, June 6-10, 2010*, pages 1155–1158, 2010.

[68] Q. Mei, C. Liu, H. Su, and C. Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006*, pages 533–542, 2006.

[69] X. Meng, F. Wei, X. Liu, M. Zhou, S. Li, and H. Wang. Entity-centric topic-oriented opinion summarization in twitter. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, pages 379–387, 2012.

[70] P. T. Metaxas, E. Mustafaraj, and D. Gayo-Avello. How (not) to predict elections. In *PASSAT/SocialCom 2011, Privacy, Security, Risk and Trust (PASSAT), 2011 IEEE Third International Conference on and 2011 IEEE Third International Conference on Social Computing (SocialCom), Boston, MA, USA, 9-11 Oct., 2011*, pages 165–171, 2011.

[71] B. Meyer, K. Bryan, Y. Santos, and B. Kim. Twitterreporter: Breaking news detection and visualization through the geo-tagged twitter network. In *Proceedings of the ISCA 26th International Conference on Computers and Their Applications, CATA 2011, March 23-15, 2011, Holiday Inn Downtown-Superdome, New Orleans, Louisiana, USA*, pages 84–89, 2011.

[72] G. A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995.

[73] S. Mukherjee, A. Malu, B. A. R., and P. Bhattacharyya. Twisent: a multistage system for analyzing sentiment in twitter. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 2531–2534, 2012.

[74] N. Naveed, T. Gottron, J. Kunegis, and A. C. Alhadi. Bad news travel fast: A content-based analysis of interestingness on twitter. In *Web Science 2011, WebSci '11, Koblenz, Germany - June 15 - 17, 2011*, pages 1–7, June 2011.

[75] NewsPoll. Two party preferred. http://polling.newspoll.com.au.tmp.anchor.net.au/image_uploads/130922%20Final%20Election%20Poll.pdf. Accessed: 2014-07-07.

[76] X. Ni, X. Quan, Z. Lu, L. Wenyin, and B. Hua. Short text clustering by finding core terms. *Knowledge and Information Systems*, 27(3):345–365, 2011.

[77] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA, May 23-26, 2010*, pages 122–129, 2010.

[78] O. Ozdikis, P. Senkul, and H. Oguztuzun. Semantic expansion of hashtags for enhanced event detection in twitter. In *International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2012, Istanbul, Turkey, 26-29 August 2012*, pages 1:1–1:6, 2012.

[79] G. Palla, I. Dernyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

[80] H.-K. Peng, J. Zhu, D. Piao, R. Yan, and Y. Zhang. Retweet modeling using conditional random fields. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on, Vancouver, BC, Canada, December 11, 2011*, pages 336–343, 2011.

[81] J. Peng, D. Yang, S.-W. Tang, J. Gao, P. yi Zhang, and Y. Fu. A concept similarity based text classification algorithm. In *Fourth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2007, 24-27 August 2007, Haikou, Hainan, China, Proceedings, Volume 1*, pages 535–539, 2007.

[82] S. Petrovic, M. Osborne, and V. Lavrenko. Rt to win! predicting message propagation in twitter. In *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*, pages 586–589, 2011.

[83] D. Pohl, A. Bouchachia, and H. Hellwagner. Automatic sub-event detection in emergency management using social media. In *Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon, France, April 16-20, 2012 (Companion Volume)*, pages 683–686, 2012.

[84] A. Rangrej, S. Kulkarni, and A. V. Tendulkar. Comparative study of clustering techniques for short text documents. In *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011 (Companion Volume)*, pages 111–112, 2011.

[85] T. Rattenbury, N. Good, and M. Naaman. Towards automatic extraction of event and place semantics from flickr tags. In *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007*, pages 103–110, 2007.

[86] ReachTel. Two party preferred result based on 2010 election distribution. `https://www.reachtel.com.au/blog/7-news-national-poll-5september13`. Accessed: 2014-07-07.

[87] C. Reed, T. Elvers, and P. Srinivasan. What's trending?: Mining topical trends in ugc systems with youtube as a case study. In *Proceedings of the Eleventh International Workshop on Multimedia Data Mining MDMKDD 2011, San Diego, California, New York, NY, USA*, pages 4:1–4:9, 2011.

[88] M. Ringsquandl and D. Petkovic. Analyzing political sentiment on twitter. In *Analyzing Microtext, Papers from the 2013 AAAI Spring Symposium, Palo Alto, California, USA, March 25-27, 2013*, pages 40–47, 2013.

[89] RoyMorgan. Two party preferred voting intention (%). `http://www.roymorgan.com/morganpoll/federal-voting/2pp-voting-intention-recent-2013-2016`. Accessed: 2014-07-07.

[90] I. Ruthven and M. Lalmas. A survey on the use of relevance feedback for information access systems. *Knowledge Engineering Review*, 18(2):95–145, June 2003.

[91] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, 2010.

[92] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[93] E. T. K. Sang and J. Bos. Predicting the 2011 dutch senate election results with twitter. In *EACL Workshop on Semantic Analysis in Social Media*, pages 53–60, 2012.

[94] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling. Twitterstand: news in tweets. In *17th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2009, November 4-6, 2009, Seattle, Washington, USA, Proceedings*, pages 42–51, 2009.

[95] A. D. Sarma, A. Jain, and C. Yu. Dynamic relationship and event discovery. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*, pages 207–216, 2011.

[96] H. Sayyadi, M. Hurst, and A. Maykov. Event detection and tracking in social streams. In *Proceedings of the Third International Conference on Weblogs and Social Media, ICWSM 2009, San Jose, California, USA, May 17-20, 2009*, pages 311–314, 2009.

[97] P. Serdyukov, V. Murdock, and R. van Zwol. Placing flickr photos on a map. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, pages 484–491, 2009.

[98] V. K. Singh, M. Gao, and R. Jain. Situation detection and control using spatio-temporal analysis of microblogs. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 1181–1182, 2010.

[99] D. H. Stern, R. Herbrich, and T. Graepel. Matchbox: large scale online bayesian recommendations. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pages 111–120, 2009.

[100] B. Suh, L. Hong, P. Pirolli, and E. H. Chi. Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In *Proceedings of the 2010 IEEE Second International Conference on Social Computing, SocialCom / IEEE International Conference on Privacy, Security, Risk and Trust, PASSAT 2010, Minneapolis, Minnesota, USA, August 20-22, 2010*, pages 177–184, 2010.

[101] E. Sun, I. Rosenn, C. Marlow, and T. M. Lento. Gesundheit! modeling contagion through facebook news feed. In *Proceedings of the Third International Conference on Weblogs and Social Media, ICWSM 2009, San Jose, California, USA, May 17-20, 2009*, pages 146–153, 2009.

[102] M. Takaffoli, F. Sangi, J. Fagnan, and O. R. Zaiane. Modec - modeling and detecting evolutions of communities. In *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*, pages 626–629, 2011.

[103] I. Taxidou. Realtime analysis of information diffusion in social media. *PVLDB*, 6(12):1416–1421, 2013.

[104] O. Tsur, D. Davidov, and A. Rappoport. Icwsm - a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA, May 23-26, 2010*, pages 162–169, 2010.

[105] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA, May 23-26, 2010*, pages 178–185, 2010.

[106] S. Unankard, L. Chen, P. Li, S. Wang, Z. Huang, M. A. Sharaf, and X. Li. On the prediction of re-tweeting activities in social networks - A report on WISE 2012 challenge. In *Web Information Systems Engineering - WISE 2012 - 13th International Conference, Paphos, Cyprus, November 28-30, 2012. Proceedings*, pages 744–754, 2012.

[107] S. Unankard, X. Li, and M. A. Sharaf. Location-based emerging event detection in social networks. In *Web Technologies and Applications - 15th Asia-Pacific Web Conference, APWeb 2013, Sydney, Australia, April 4-6, 2013. Proceedings*, pages 280–291, 2013.

[108] S. Unankard, X. Li, and M. A. Sharaf. Emerging event detection in social networks with location sensitivity. *World Wide Web*, pages 1–25, 2014.

[109] S. Unankard, X. Li, M. A. Sharaf, J. Zhong, and X. Li. Predicting elections from social networks based on sub-event detection and sentiment analysis. In *Web Information Systems Engineering - WISE 2014 - 15th International Conference, Thessaloniki, Greece, October 12-14, 2014, Proceedings, Part II*, pages 1–16, 2014.

[110] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan. A system for real-time twitter sentiment analysis of 2012 u.s. presidential election cycle. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the System Demonstrations, July 10, 2012, Jeju Island, Korea*, pages 115–120, 2012.

[111] J. Wang, Y. Zhou, L. Li, B. Hu, and X. Hu. Improving short text clustering performance with keyword expansion. In *The Sixth International Symposium on Neural Networks, ISNN 2009, Wuhan, China, May 26-29, 2009, Proceedings, Part IV*, pages 291–298, 2009.

[112] K. Watanabe, M. Ochi, M. Okabe, and R. Onai. Jasmine: a real-time local-event detection system based on geolocation information propagated to microblogs. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, pages 2541–2544, 2011.

[113] A. Weiler, M. Grossniklaus, and M. H. Scholl. Event identification and tracking in social media streaming data. In *Proceedings of the Workshops of the EDBT/ICDT 2014 Joint Conference (EDBT/ICDT 2014), Athens, Greece, March 28, 2014.*, pages 282–287, 2014.

[114] J. Weng and B.-S. Lee. Event detection in twitter. In *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*, pages 401–408, 2011.

[115] C. Wilson, B. Boe, A. Sala, K. P. N. Puttaswamy, and B. Y. Zhao. User interactions in social networks and their implications. In *Proceedings of the 2009 EuroSys Conference, Nuremberg, Germany, April 1-3, 2009*, pages 205–218, 2009.

[116] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6-8 October 2005, Vancouver, British Columbia, Canada*, pages 347–354, 2005.

[117] J. Yang and J. Leskovec. Modeling information diffusion in implicit networks. In *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*, pages 599–608, 2010.

[118] Y. Yang, T. Pierce, and J. G. Carbonell. A study of retrospective and on-line event detection. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia*, pages 28–36, 1998.

[119] Z. Yang, J. Guo, K. Cai, J. Tang, J. Li, L. Zhang, and Z. Su. Understanding retweeting behaviors in social networks. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 1633–1636, 2010.

[120] Z. Yang, J. Guo, K. Cai, J. Tang, J. Li, L. Zhang, and Z. Su. Understanding retweeting behaviors in social networks. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 1633–1636, 2010.

[121] T. R. Zaman, R. Herbrich, and D. H. Stern. Predicting information spreading in twitter. *Computational Social Science and the Wisdom of Crowds*, 55:1–4, 2010.

[122] D. Zhao and M. B. Rosson. How and why people twitter: the role that micro-blogging plays in informal communication at work. In *Proceedings of the 2009 International ACM SIGGROUP*

Conference on Supporting Group Work, GROUP 2009, Sanibel Island, Florida, USA, May 10-13, 2009, pages 243–252, 2009.

[123] Q. Zhao and P. Mitra. Event detection and visualization for social text streams. In *Proceedings of the First International Conference on Weblogs and Social Media, ICWSM 2007, Boulder, Colorado, USA, March 26-28, 2007*, pages 1–4, 2007.

Appendix

Awards During Candidature

- Xue Li, Weitong Chen, Sayan Unankard, Ling Chen, Xiaofang Zhou, Shazia Sadiq. **The Best Use of Open Data Award**. WikiQueensland- a big data fusion system, In *The 2014 Premier's Awards for Open Data*, Queensland, Australia, 2014.

- Xue Li, Weitong Chen, Sayan Unankard, Ling Chen, Xiaofang Zhou, Shazia Sadiq. **Microsoft StartUp Q Award**. WikiQueensland- a big data fusion system, In *The 2014 Premier's Awards for Open Data*, Queensland, Australia, 2014.

- Sayan Unankard. **Runner-up** of the School of Information Technology and Electrical Engineering, The University of Queensland. In *Three Minute Thesis (3MT) competition*, 2014.

- Sayan Unankard, Xue Li and Mohamed A. Sharaf. **Best Research Poster Awards - Joint First Prize**. Sub-Event Detection and Sentiment Analysis in Social Networks, In *The 2014 Australasia Database Conference PhD School in Big Data*, 2014.

- Sayan Unankard, Xue Li and Mohamed A. Sharaf. **Best Student Paper Award**. Location-based Emerging Event Detection in Social Networks. In *Asia-Pacific Web Conference*, 2013.

- Sayan Unankard, Ling Chen, Peng Li, Sen Wang, Zi Huang, Mohamed A. Sharaf, Xue Li. **Champion Data mining track**. On the Prediction of Re-tweeting Activities in Social Networks A Report on WISE 2012 Challenge. In *Web Information Systems Engineering Conference*, 2012.